US-PAT-NO: 5629778

DOCUMENT-IDENTIFIER: US 5629778 A

TITLE: Method and apparatus for reduction of image data compression noise

DATE-ISSUED: May 13, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Reuman; Steven R.	Acton	MA	N/A	N/A

US-CL-CURRENT: 382/252

ABSTRACT:

An image encoding/decoding apparatus for performing transform coding by a method in which blocking artifacts are suppressed or eliminated is disclosed in which encoded data, transmitted by the encoding apparatus, is converted into received image data terms which are subsequently overlap transformed into frequency coefficients for modification by means of a filtering operation utilizing a quantization error matrix. The quantization error matrix can be derived from quantization error data generated in the encoding unit, or can be provided as a look-up table in the decoding unit. The modified frequency coefficients are converted into reduced-noise image data terms for

reconstruction into a digital image.

52 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 5

---------- KWIC ---------

Claims Text - CLTX (14):
converting said **modified coefficient matrices** into filtered **coefficient** matrices by means of said quantization error **matrix and said filter parameters,** **said modified coefficient matrix** comprising terms denoted by Sf.sub.s,r (.upsilon.,.mu.), said step of converting said **modified coefficient matrices** performed in accordance with the equation, ##EQU19## transforming said filtered **coefficient** matrices into filtered image-data matrices in accordance with the **transform** equation,

Claims Text - CLTX (40):
transforming said overlapped image-data **matrices into modified coefficient matrices** comprising terms denoted by Sv.sub.s,r (.upsilon.,.mu.), by means of an orthogonal **transform** basis matrix C in accordance with the matrix equation,

SEARCH RESULTS [PDF Full-Text (1200 KB)] PREVIOUS NEXT
DOWNLOAD CITATION

A new, fast, and efficient image codec based on set partitioning in hierarchical trees

Said, A. Pearlman, W.A.
Fac. of Electr. Eng., State Univ. of Campinas;

This paper appears in: **Circuits and Systems for Video Technology, IEEE Transactions on**

Publication Date: Jun 1996
On page(s): 243-250
Volume: 6, Issue: 3
ISSN: 1051-8215
References Cited: 15
CODEN: ITCTEM
INSPEC Accession Number: 5295430

Abstract:
Embedded zerotree wavelet (EZW) coding, introduced by Shapiro (see IEEE Trans. Signal Processing, vol.41, no.12, p.3445, 1993), is a very effective and computationally simple technique for image compression. We offer an alternative explanation of the principles of its operation, so that the reasons for its excellent performance can be better understood. These principles are partial ordering by magnitude with a set partitioning sorting algorithm, ordered bit plane transmission, and exploitation of self-similarity across different scales of an image wavelet transform. Moreover, we present a new and different implementation based on set partitioning in hierarchical trees (SPIHT), which provides even better performance than our previously reported extension of EZW that surpassed the performance of the original EZW. The image coding results, calculated from actual file sizes and images reconstructed by the decoding algorithm, are either comparable to or surpass previous results obtained through much more sophisticated and computationally complex methods. In addition, the new coding and decoding procedures are extremely fast, and they can be made even faster, with only small loss in performance, by omitting entropy coding of the bit stream by the arithmetic code

Index Terms:
arithmetic codes codecs data compression entropy codes image coding image reconstruction transform coding trees (mathematics) wavelet transforms arithmetic code decoding decoding algorithm

embedded zerotree wavelet coding entropy coding file sizes image codec image coding image compression image reconstruction image wavelet transform ordered bit plane transmission partial ordering performance self-similarity set partitioning in hierarchical trees set partitioning sorting algorithm

Documents that cite this document

Select link to view other documents in the database that cite this one.

Reference list:

1. J. M.Shapiro, &ldquo;Embedded image coding using zerotrees of wavelets coefficients,&rdquo; *IEEE Trans. Signal Processing,* vol. 41, pp. 3445-3462, Dec. 1993.
[Abstract] [PDF Full-Text (2344KB)]

2. M.Rabbani and P. W.Jones, *Digital Image Compression Techniques.* Bellingham, WA: SPIE Opt. Eng. Press, 1991.

3. R. A.DeVore, B.Jawerth, and B. J.Lucier, &ldquo;Image compression through wavelet transform coding,&rdquo; *IEEE Trans. Inform. Theory*, vol. 38, pp. 719-746, Mar. 1992.
[Abstract] [PDF Full-Text (2876KB)]

4. E. H.Adelson, E.Simoncelli, and R.Hingorani, &ldquo;Orthogonal pyramid transforms for image coding,&rdquo; *Proc. SPIE* Cambridge, MA, vol. 845, Visual Commun. and Image Proc. II, pp. 50-58, Oct. 1987.

5. M.Antonini, M.Barlaud, P.Mathieu, and I.Daubechies, &ldquo;Image coding using wavelet transform,&rdquo; *IEEE Trans. Image Processing*, vol. 1, pp. 205-220, Apr. 1992.
[Abstract] [PDF Full-Text (1788KB)]

6. A.Said and W. A.Pearlman, &ldquo;Image compression using the spatial-orientation tree,&rdquo; *IEEE Int. Symp. Circuits and Systems* Chicago, IL, pp. 279-282, May 1993.
[Abstract] [PDF Full-Text (466KB)]

7. I. H.Witten, R. M.Neal, and J. G.Cleary, &ldquo;Arithmetic coding for data compression,&rdquo; *Commun. ACM*, vol. 30, pp. 520-540, June 1987.
[CrossRef]

8. P.Sriram and M. W.Marcellin, &ldquo;Wavelet coding of images using trellis coded quantization,&rdquo; *SPIE Conf. Visual Inform. Process.* Orlando, FL, pp. 238-247, Apr. 1992.

9. Y. H.Kim and J. W.Modestino, &ldquo;Adaptive entropy coded subband coding of images,&rdquo; *IEEE Trans. Image Processing*, vol. 1, pp. 31-48, Jan. 1992.
[Abstract] [PDF Full-Text (1896KB)]

10. N.Tanabe and N.Farvardin, &ldquo;Subband image coding using entropy-constrained quantization over noisy channels,&rdquo; *IEEE J. Select. Areas Commun.*, vol. 10, pp. 926-943, June 1992.
[Abstract] [PDF Full-Text (1796KB)]

11. R. L.Joshi, V. J.Crump, and T. R.Fischer, &ldquo;Image

subband coding using arithmetic and trellis coded quantization,&rdquo; *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 515-523, Dec. 1995.
[Abstract] [PDF Full-Text (1092KB)]

12. R. L.Joshi, T. R.Fischer, and R. H.Bamberger, &ldquo;Optimum classification in subband coding of images,&rdquo; *Proc. 1994 IEEE Int. Conf. Image Processing* Austin, TX, vol. II, pp. 883-887, Nov. 1994.
[Abstract] [PDF Full-Text (320KB)]

13. J. H.Kasner and M. W.Marcellin, &ldquo;Adaptive wavelet coding of images,&rdquo; *Proc. 1994 IEEE Conf. Image Processing* Austin, TX, vol. 3, pp. 358-362, Nov. 1994.
[Abstract] [PDF Full-Text (396KB)]

14. A.Said and W. A.Pearlman, &ldquo;Reversible image compression via multiresolution representation and predictive coding,&rdquo; *Proc. SPIE Conf. Visual Communications and Image Processing '93*, Proc. SPIE 2094 Cambridge, MA, pp. 664-674, Nov. 1993.

15. D. P.de Garrido, W. A.Pearlman, and W. A.Finamore, &ldquo;A clustering algorithm for entropy-constrained vector quantizer design with applications in coding image pyramids,&rdquo; *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 83-95, Apr. 1995.
[Abstract] [PDF Full-Text (1308KB)]

SEARCH RESULTS [PDF Full-Text (1200 KB)] PREVIOUS NEXT
DOWNLOAD CITATION

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback |Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms| Back to Top

A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees

Amir Said, *Member, IEEE,* and William A. Pearlman, *Senior Member, IEEE*

***Abstract*— Embedded zerotree wavelet (EZW) coding, introduced by J. M. Shapiro, is a very effective and computationally simple technique for image compression. Here we offer an alternative explanation of the principles of its operation, so that the reasons for its excellent performance can be better understood. These principles are partial ordering by magnitude with a set partitioning sorting algorithm, ordered bit plane transmission, and exploitation of self-similarity across different scales of an image wavelet transform. Moreover, we present a new and different implementation based on set partitioning in hierarchical trees (SPIHT), which provides even better performance than our previously reported extension of EZW that surpassed the performance of the original EZW. The image coding results, calculated from actual file sizes and images reconstructed by the decoding algorithm, are either comparable to or surpass previous results obtained through much more sophisticated and computationally complex methods. In addition, the new coding and decoding procedures are extremely fast, and they can be made even faster, with only small loss in performance, by omitting entropy coding of the bit stream by arithmetic code.**

I. Introduction

IMAGE compression techniques, especially nonreversible or lossy ones, have been known to grow computationally more complex as they grow more efficient, confirming the tenets of source coding theorems in information theory that a code for a (stationary) source approaches optimality in the limit of infinite computation (source length). Notwithstanding, the image coding technique called embedded zerotree wavelet (EZW), introduced by Shapiro [1], interrupted the simultaneous progression of efficiency and complexity. This technique not only was competitive in performance with the most complex techniques, but was extremely fast in execution and produced an embedded bit stream. With an embedded bit stream, the reception of code bits can be stopped at any point and the image can be decompressed and reconstructed. Following that significant work, we developed an alternative exposition of the underlying principles of the EZW technique and presented an extension that achieved even better results [6].

In this article, we again explain that the EZW technique is based on three concepts: 1) partial ordering of the transformed image elements by magnitude, with transmission of order by a subset partitioning algorithm that is duplicated at the decoder, 2) ordered bit plane transmission of refinement bits, and 3) exploitation of the self-similarity of the image wavelet transform across different scales. As to be explained, the partial ordering is a result of comparison of transform element (coefficient) magnitudes to a set of octavely decreasing thresholds. We say that an element is significant or insignificant with respect to a given threshold, depending on whether or not it exceeds that threshold.

In this work, crucial parts of the coding process—the way subsets of coefficients are partitioned and how the significance information is conveyed—are fundamentally different from the aforementioned works. In the previous works, arithmetic coding of the bit streams was essential to compress the ordering information as conveyed by the results of the significance tests. Here, the subset partitioning is so effective and the significance information so compact that even binary uncoded transmission achieves about the same or better performance than in these previous works. Moreover, the utilization of arithmetic coding can reduce the mean squared error or increase the peak signal-to-noise ratio (PSNR) by 0.3–0.6 dB for the same rate or compressed file size and achieve results which are equal to or superior to any previously reported, regardless of complexity. Execution times are also reported to indicate the rapid speed of the encoding and decoding algorithms. The transmitted code or compressed image file is completely embedded, so that a single file for an image at a given code rate can be truncated at various points and decoded to give a series of reconstructed images at lower rates. Previous versions [1], [6] could not give their best performance with a single embedded file and required, for each rate, the optimization of a certain parameter. The new method solves this problem by changing the transmission priority and yields, with one embedded file, its top performance for all rates.

The encoding algorithms can be stopped at any compressed file size or let run until the compressed file is a representation of a *nearly* lossless image. We say *nearly* lossless because the compression *may* not be reversible, as the wavelet transform filters, chosen for lossy coding, have noninteger tap weights and produce noninteger transform coefficients, which are truncated to finite precision. For perfectly reversible compression, one must use an integer multiresolution transform, such as the S+P transform introduced in [14], which yields excellent reversible compression results when used with the new extended EZW techniques.

Manuscript received April 18, 1995. This paper was recommended by Associate Editor C. Gonzales. This paper was presented in part at the IEEE International Symposium on Circuits and Systems, Chicago, IL, May 1993.

A. Said is with the Faculty of Electrical Engineering, State University of Campinas (UNICAMP), Campinas, SP, 13081, Brazil.

W. A. Pearlman is with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA.

Publisher Item Identifier S 1051-8215(96)04105-5.

This paper is organized as follows. The next section, Section II, describes an embedded coding or progressive transmission scheme that prioritizes the code bits according to their reduction in distortion. Section III explains the principles of partial ordering by coefficient magnitude and ordered bit plane transmission, which suggest a basis for an efficient coding method. The set partitioning sorting procedure and spatial orientation trees (called zerotrees previously) are detailed in Sections IV and V, respectively. Using the principles set forth in the previous sections, the coding and decoding algorithms are fully described in Section VI. In Section VII, rate, distortion, and execution time results are reported on the operation of the coding algorithm on test images and the decoding algorithm on the resultant compressed files. The figures on rate are calculated from actual compressed file sizes and on mean squared error or PSNR from the reconstructed images given by the decoding algorithm. Some reconstructed images are also displayed. These results are put into perspective by comparison to previous work. The conclusion of the paper is in Section VIII.

II. Progressive Image Transmission

We assume that the original image is defined by a set of pixel values $p_{i,j}$, where (i, j) is the pixel coordinate. To simplify the notation we represent two-dimensional (2-D) arrays with bold letters. The coding is actually done to the array

$$\mathbf{c} = \Omega(\mathbf{p}) \tag{1}$$

where $\Omega(\cdot)$ represents a unitary hierarchical subband transformation (e.g., [4]). The 2-D array $\mathbf{c}$ has the same dimensions of $\mathbf{p}$, and each element $c_{i,j}$ is called *transform coefficient* at coordinate (i, j). For the purpose of coding, we assume that each $c_{i,j}$ is represented with a fixed-point binary format, with a small number of bits—typically 16 or less—and can be treated as an integer.

In a progressive transmission scheme, the decoder initially sets the reconstruction vector $\hat{\mathbf{c}}$ to zero and updates its components according to the coded message. After receiving the value (approximate or exact) of some coefficients, the decoder can obtain a reconstructed image

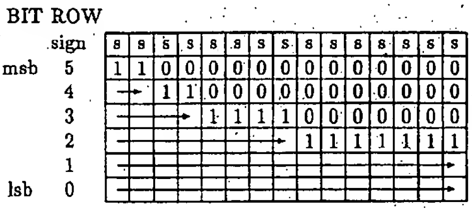
$$\hat{\mathbf{p}} = \Omega^{-1}(\hat{\mathbf{c}}). \tag{2}$$

A major objective in a progressive transmission scheme is to select the most important information—which yields the largest distortion reduction—to be transmitted first. For this selection, we use the mean squared-error (MSE) distortion measure

$$D_{\text{mse}}(\mathbf{p} - \hat{\mathbf{p}}) = \frac{\|\mathbf{p} - \hat{\mathbf{p}}\|^2}{N} = \frac{1}{N} \sum_i \sum_j (p_{i,j} - \hat{p}_{i,j})^2 \tag{3}$$

where N is the number of image pixels. Furthermore, we use the fact that the Euclidean norm is invariant to the unitary transformation Ω, i.e.,

$$D_{\text{mse}}(\mathbf{p} - \hat{\mathbf{p}}) = D_{\text{mse}}(\mathbf{c} - \hat{\mathbf{c}}) = \frac{1}{N} \sum_i \sum_j (c_{i,j} - \hat{c}_{i,j})^2. \tag{4}$$



BIT ROW

	sign	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s
msb	5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	4		→	1	1	0	0	0	0	0	0	0	0	0	0	0
	3				→	1	1	1	1	0	0	0	0	0	0	0
	2								→	1	1	1	1	1	1	1
	1															→
lsb	0															→

Fig. 1. Binary representation of the magnitude-ordered coefficients.

From (4) it is clear that if the exact value of the transform coefficient $c_{i,j}$ is sent to the decoder, then the MSE decreases by $|c_{i,j}|^2/N$. This means that the coefficients with larger magnitude should be transmitted first because they have a larger content of information.[1] This corresponds to the progressive transmission method proposed by DeVore *et al.* [3]. Extending their approach, we can see that the information in the value of $|c_{i,j}|$ can also be ranked according to its binary representation, and the most significant bits should be transmitted first. This idea is used, for example, in the bit-plane method for progressive transmission [2].

Following, we present a progressive transmission scheme that incorporates these two concepts: ordering the coefficients by magnitude and transmitting the most significant bits first. To simplify the exposition, we first assume that the ordering information is explicitly transmitted to the decoder. Later, we show a much more efficient method to code the ordering information.

III. Transmission of the Coefficient Values

Let us assume that the coefficients are ordered according to the minimum number of bits required for its magnitude binary representation, that is, ordered according to a one-to-one mapping $\eta : I \mapsto I^2$, such that

$$\lfloor \log_2 |c_{\eta(k)}| \rfloor \geq \lfloor \log_2 |c_{\eta(k+1)}| \rfloor, \quad k = 1, \cdots, N. \tag{5}$$

Fig. 1 shows the schematic binary representation of a list of magnitude-ordered coefficients. Each column k in Fig. 1 contains the bits of $c_{\eta(k)}$. The bits in the top row indicate the sign of the coefficient. The rows are numbered from the bottom up, and the bits in the lowest row are the least significant.

Now, let us assume that, besides the ordering information, the decoder also receives the numbers μ_n corresponding to the number of coefficients such that $2^n \leq |c_{i,j}| < 2^{n+1}$. In the example of Fig. 1 we have $\mu_5 = 2$, $\mu_4 = 2$, $\mu_3 = 4$, etc. Since the transformation Ω is unitary, all bits in a row have the same content of information, and the most effective order for progressive transmission is to sequentially send the bits in each row, as indicated by the arrows in Fig. 1. Note that,

[1] Here the term *information* is used to indicate how much the distortion can decrease after receiving that part of the coded message.

because the coefficients are in decreasing order of magnitude, the leading "0" bits and the first "1" of any column do not need to be transmitted, since they can be inferred from μ_n and the ordering.

The progressive transmission method outlined above can be implemented with the following algorithm to be used by the encoder.

Algorithm I:

1) output $n = \lfloor \log_2(\max_{(i,j)}\{|c_{i,j}|\})\rfloor$ to the decoder;
2) output μ_n, followed by the pixel coordinates $\eta(k)$ and sign of each of the μ_n coefficients such that $2^n \leq |c_{\eta(k)}| < 2^{n+1}$ **(sorting pass)**;
3) output the nth most significant bit of all the coefficients with $|c_{i,j}| \geq 2^{n+1}$ (i.e., those that had their coordinates transmitted in previous sorting passes), in the same order used to send the coordinates **(refinement pass)**;
4) decrement n by one, and go to Step **2**).

The algorithm stops at the desired rate or distortion. Normally, good quality images can be recovered after a relatively small fraction of the pixel coordinates are transmitted.

The fact that this coding algorithm uses uniform scalar quantization may give the impression that it must be much inferior to other methods that use nonuniform and/or vector quantization. However, this is not the case: the ordering information makes this simple quantization method very efficient. On the other hand, a large fraction of the "bit-budget" is spent in the sorting pass, and it is there that the sophisticated coding methods are needed.

IV. Set Partitioning Sorting Algorithm

One of the main features of the proposed coding method is that the ordering data is not explicitly transmitted. Instead, it is based on the fact that the execution path of any algorithm is defined by the results of the comparisons on its branching points. So, if the encoder and decoder have the same sorting algorithm, then the decoder can duplicate the encoder's execution path if it receives the results of the magnitude comparisons, and the ordering information can be recovered from the execution path.

One important fact used in the design of the sorting algorithm is that we do not need to sort all coefficients. Actually, we need an algorithm that simply selects the coefficients such that $2^n \leq |c_{i,j}| < 2^{n+1}$, with n decremented in each pass. Given n, if $|c_{i,j}| \geq 2^n$ then we say that a coefficient is *significant;* otherwise it is called *insignificant.*

The sorting algorithm divides the set of pixels into partitioning subsets $\mathcal{T}_m$ and performs the magnitude test

$$\max_{(i,j)\in\mathcal{T}_m} \{|c_{i,j}|\} \geq 2^n? \qquad (6)$$

If the decoder receives a "no" to that answer (the *subset* is insignificant), then it knows that all coefficients in $\mathcal{T}_m$ are insignificant. If the answer is "yes" (the subset is significant), then a certain rule shared by the encoder and the decoder is used to partition $\mathcal{T}_m$ into new subsets $\mathcal{T}_{m,l}$, and the significance test is then applied to the new subsets. This set division process continues until the magnitude test is done to all single coordinate significant subsets in order to identify each significant coefficient.

To reduce the number of magnitude comparisons (message bits) we define a set partitioning rule that uses an expected ordering in the hierarchy defined by the subband pyramid. The objective is to create new partitions such that subsets expected to be insignificant contain a large number of elements, and subsets expected to be significant contain only one element.

To make clear the relationship between magnitude comparisons and message bits, we use the function

$$S_n(\mathcal{T}) = \begin{cases} 1, & \max_{(i,j)\in\mathcal{T}} \{|c_{i,j}|\} \geq 2^n, \\ 0, & \text{otherwise} \end{cases} \qquad (7)$$

to indicate the significance of a set of coordinates $\mathcal{T}$. To simplify the notation of single pixel sets, we write $S_n(\{(i,j)\})$ as $S_n(i,j)$.

V. Spatial Orientation Trees

Normally, most of an image's energy is concentrated in the low frequency components. Consequently, the variance decreases as we move from the highest to the lowest levels of the subband pyramid. Furthermore, it has been observed that there is a spatial self-similarity between subbands, and the coefficients are expected to be better magnitude-ordered if we move downward in the pyramid following the same spatial orientation. [Note the mild requirements for ordering in (5).] For instance, large low-activity areas are expected to be identified in the highest levels of the pyramid, and they are replicated in the lower levels at the same spatial locations.

A tree structure, called *spatial orientation tree,* naturally defines the spatial relationship on the hierarchical pyramid. Fig. 2 shows how our spatial orientation tree is defined in a pyramid constructed with recursive four-subband splitting. Each node of the tree corresponds to a pixel and is identified by the pixel coordinate. Its direct descendants (offspring) correspond to the pixels of the same spatial orientation in the next finer level of the pyramid. The tree is defined in such a way that each node has either no offspring (the leaves) or four offspring, which always form a group of 2 × 2 adjacent pixels. In Fig. 2, the arrows are oriented from the parent node to its four offspring. The pixels in the highest level of the pyramid are the tree roots and are also grouped in 2 × 2 adjacent pixels. However, their offspring branching rule is different, and in each group, one of them (indicated by the star in Fig. 2) has no descendants.

The following sets of coordinates are used to present the new coding method:

- $\mathcal{O}(i,j)$: set of coordinates of all offspring of node (i,j);
- $\mathcal{D}(i,j)$: set of coordinates of all descendants of the node (i,j);
- $\mathcal{H}$: set of coordinates of all spatial orientation tree roots (nodes in the highest pyramid level);
- $\mathcal{L}(i,j) = \mathcal{D}(i,j) - \mathcal{O}(i,j)$.

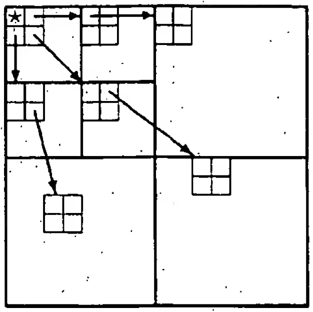


Fig. 2. Examples of parent-offspring dependencies in the spatial-orientation tree.

For instance, except at the highest and lowest pyramid levels, we have

$$\mathcal{O}(i, j) = \{(2i, 2j), (2i, 2j+1), (2i+1, 2j), (2i+1, 2j+1)\}. \quad (8)$$

We use parts of the spatial orientation trees as the partitioning subsets in the sorting algorithm. The set partitioning rules are simply the following.

1) The initial partition is formed with the sets $\{(i, j)\}$ and $\mathcal{D}(i, j)$, for all $(i, j) \in \mathcal{H}$.
2) If $\mathcal{D}(i, j)$ is significant, then it is partitioned into $\mathcal{L}(i, j)$ plus the four single-element sets with $(k, l) \in \mathcal{O}(i, j)$.
3) If $\mathcal{L}(i, j)$ is significant, then it is partitioned into the four sets $\mathcal{D}(k, l)$, with $(k, l) \in \mathcal{O}(i, j)$.

VI. Coding Algorithm

Since the order in which the subsets are tested for significance is important, in a practical implementation the significance information is stored in three ordered lists, called list of insignificant sets (LIS), list of insignificant pixels (LIP), and list of significant pixels (LSP). In all lists each entry is identified by a coordinate (i, j), which in the LIP and LSP represents individual pixels, and in the LIS represents either the set $\mathcal{D}(i, j)$ or $\mathcal{L}(i, j)$. To differentiate between them, we say that a LIS entry is of type A if it represents $\mathcal{D}(i, j)$, and of type B if it represents $\mathcal{L}(i, j)$.

During the sorting pass (see Algorithm I), the pixels in the LIP—which were insignificant in the previous pass—are tested, and those that become significant are moved to the LSP. Similarly, sets are sequentially evaluated following the LIS order, and when a set is found to be significant it is removed from the list and partitioned. The new subsets with more than one element are added back to the LIS, while the single-coordinate sets are added to the end of the LIP or the LSP, depending whether they are insignificant or significant, respectively. The LSP contains the coordinates of the pixels that are visited in the refinement pass.

Below we present the new encoding algorithm in its entirety. It is essentially equal to Algorithm I, but uses the set-partitioning approach in its sorting pass.

Algorithm II:

```
1) Initialization: output $n = \lfloor \log_2(\max_{(i,j)}\{|c_{i,j}|\}) \rfloor$;
   set the LSP as an empty list, and add the coordinates
   $(i, j) \in \mathcal{H}$ to the LIP, and only those with descendants
   also to the LIS, as type $A$ entries.
2) Sorting Pass:
   2.1) for each entry $(i, j)$ in the LIP do:
        2.1.1) output $S_n(i, j)$;
        2.1.2) if $S_n(i, j) = 1$ then move $(i, j)$ to the LSP
               and output the sign of $c_{i,j}$;
   2.2) for each entry $(i, j)$ in the LIS do:
        2.2.1) if the entry is of type $A$ then
               • output $S_n(\mathcal{D}(i, j))$;
               • if $S_n(\mathcal{D}(i, j)) = 1$ then
                 * for each $(k, l) \in \mathcal{O}(i, j)$ do:
                   • output $S_n(k, l)$;
                   • if $S_n(k, l) = 1$ then add $(k, l)$ to the
                     LSP and output the sign of $c_{k,l}$;
                   • if $S_n(k, l) = 0$ then add $(k, l)$ to the
                     end of the LIP;
                 * if $\mathcal{L}(i, j) \neq \emptyset$ then move $(i, j)$ to the
                   end of the LIS, as an entry of type $B$,
                   and go to Step 2.2.2); otherwise, remove
                   entry $(i, j)$ from the LIS;
        2.2.2) if the entry is of type $B$ then
               • output $S_n(\mathcal{L}(i, j))$;
               • if $S_n(\mathcal{L}(i, j)) = 1$ then
                 * add each $(k, l) \in \mathcal{O}(i, j)$ to the end of
                   the LIS as an entry of type $A$;
                 * remove $(i, j)$ from the LIS.
3) Refinement Pass: for each entry $(i, j)$ in the LSP,
   except those included in the last sorting pass (i.e., with
   same $n$), output the $n$th most significant bit of $|c_{i,j}|$;
4) Quantization-Step Update: decrement $n$ by 1 and go
   to Step 2.
```

One important characteristic of the algorithm is that the entries added to the end of the LIS in Step 2.2) are evaluated before that same sorting pass ends. So, when we say "for each entry in the LIS" we also mean those that are being added to its end. With Algorithm II, the rate can be precisely controlled because the transmitted information is formed of single bits. The encoder can also use the property in (4) to estimate the progressive distortion reduction and stop at a desired distortion value.

Note that in Algorithm II, all branching conditions based on the significance data S_n—which can only be calculated with the knowledge of $c_{i,j}$—are output by the encoder. Thus, to obtain the desired decoder's algorithm, which duplicates the encoder's execution path as it sorts the significant coefficients, we simply have to replace the words *output* by *input* in Algorithm II. Comparing the algorithm above to Algorithm I, we can see that the ordering information $\eta(k)$ is recovered when the coordinates of the significant coefficients are added to the end of the LSP; that is, the coefficients pointed by the coordinates in the LSP are sorted as in (5). But note that

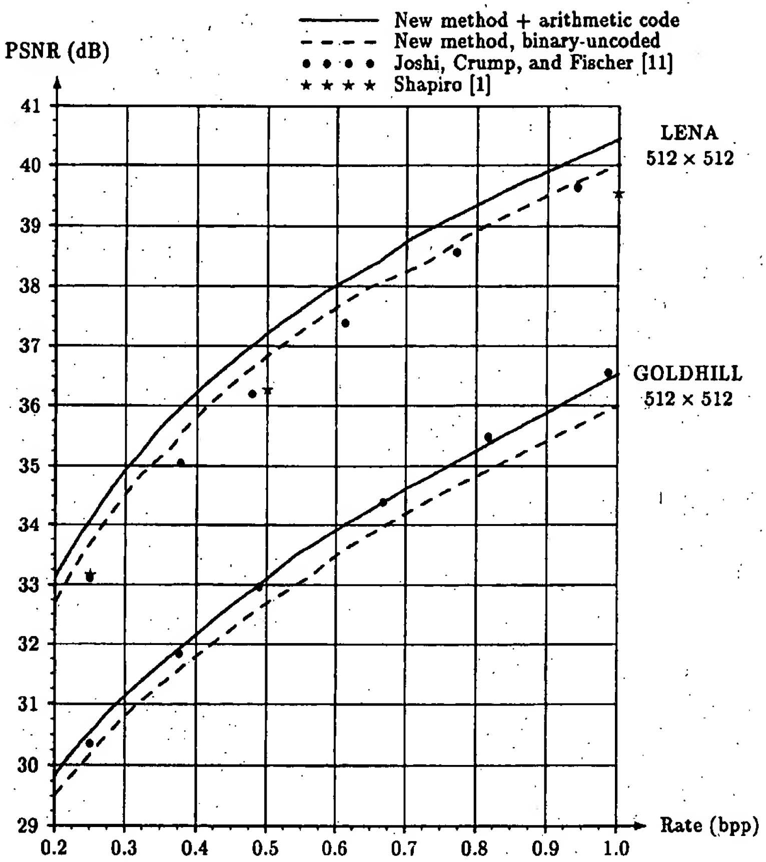


Fig. 3. Comparative evaluation of the new coding method.

whenever the decoder inputs data, its three control lists (LIS, LIP, and LSP) are identical to the ones used by the encoder at the moment it outputs that data, which means that the decoder indeed recovers the ordering from the execution path. It is easy to see that with this scheme, coding and decoding have the same computational complexity.

An additional task done by decoder is to update the reconstructed image. For the value of n when a coordinate is moved to the LSP, it is known that $2^n \leq |c_{i,j}| < 2^{n+1}$. So, the decoder uses that information, plus the sign bit that is input just after the insertion in the LSP, to set $\hat{c}_{i,j} = \pm 1.5 \times 2^n$. Similarly, during the refinement pass, the decoder adds or subtracts 2^{n-1} to $\hat{c}_{i,j}$ when it inputs the bits of the binary representation of $|c_{i,j}|$. In this manner, the distortion gradually decreases during both the sorting and refinement passes.

As with any other coding method, the efficiency of Algorithm II can be improved by entropy-coding its output, but at the expense of a larger coding/decoding time. Practical experiments have shown that normally there is little to be gained by entropy-coding the coefficient signs or the bits put out during the refinement pass. On the other hand, the significance values are not equally probable, and there is a statistical dependence between $S_n(i, j)$ and $S_n[\mathcal{D}(i, j)]$ and also between the significance of adjacent pixels.

We exploited this dependence using the adaptive arithmetic coding algorithm of Witten *et al.* [7]. To increase the coding efficiency, groups of 2 × 2 coordinates were kept together in the lists, and their significance values were coded as a single symbol by the arithmetic coding algorithm. Since the decoder only needs to know the transition from insignificant to significant (the inverse is impossible), the amount of information that needs to be coded changes according to the number m of insignificant pixels in that group, and in each case it can be conveyed by an entropy-coding alphabet with 2^m symbols. With arithmetic coding it is straightforward to use several adaptive models [7], each with 2^m symbols, $m \in \{1, 2, 3, 4\}$, to code the information in a group of four pixels.

By coding the significance information together, the average bit rate corresponds to an mth order entropy. At the same time, by using different models for the different number of insignificant pixels, each adaptive model contains probabilities *conditioned* to the fact that a certain number of adjacent pixels are significant or insignificant. This way the dependence between magnitudes of adjacent pixels is fully exploited. The scheme above was also used to code the significance of trees rooted in groups of 2 × 2 pixels.

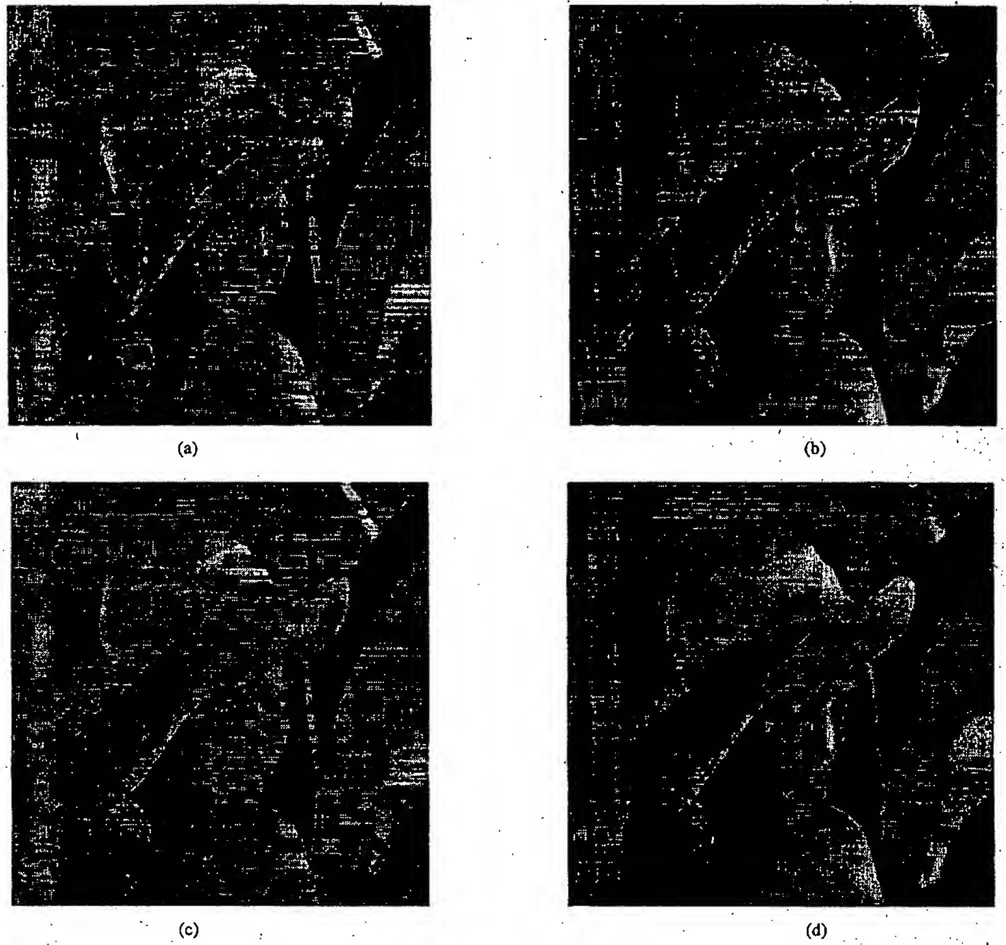


Fig. 4. Images obtained with the arithmetic code version of the new coding method. (a) Original LENA, (b) rate = 0.5 bpp, PSNR = 37.2 dB, (c) rate = 0.25 bpp, PSNR = 34.1 dB, (d) rate = 0.15 bpp, PSNR = 31.9 dB.

TABLE I
EFFECT OF ENTROPY-CODING THE SIGNIFICANCE INFORMATION ON THE CPU TIMES (s) TO CODE AND DECODE THE IMAGE LENA 512 × 512 (IBM RS/6000 WORKSTATION)

rate (bpp)	binary uncoded		arithmetic coded	
	code	decode	code	decode
0.25	0.07	0.04	0.18	0.14
0.50	0.14	0.09	0.33	0.29
1.00	0.27	0.17	0.64	0.57

With arithmetic entropy-coding it is still possible to produce a coded file with the exact code rate and possibly a few unused bits to pad the file to the desired size.

VII. NUMERICAL RESULTS

The following results were obtained with monochrome, 8 bpp, 512 × 512 images. Practical tests have shown that the pyramid transformation does not have to be exactly unitary, so we used five-level pyramids constructed with the 9/7-tap filters of [5], and using a "reflection" extension at the image edges. It is important to observe that the bit rates are not entropy estimates—they were calculated from the actual size of the compressed files. Furthermore, by using the progressive transmission ability, the sets of distortions are obtained from *the same file,* that is, the decoder read the first bytes of the file (up to the desired rate), calculated the inverse subband transformation, and then compared the recovered image with the original. The distortion is measured by the peak signal to noise ratio

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right)\mathrm{dB} \tag{9}$$

where MSE denotes the mean squared-error between the original and reconstructed images.

Results are obtained both with and without entropy-coding the bits put out with Algorithm II. We call the version without entropy coding *binary-uncoded.* In Fig. 3 are plotted the PSNR versus rate obtained for the luminance (Y) component of LENA both for binary uncoded and entropy-coded using arithmetic code. Also in Fig. 3, the same is plotted for the luminance image GOLDHILL. The numerical results with arithmetic coding surpass in almost all respects the best efforts previously reported, despite their sophisticated and computationally complex algorithms (e.g., [5], [8]–[10], [13], [15]). Even the numbers obtained with the binary uncoded versions are superior to those in all these schemes, except possibly the arithmetic and entropy constrained trellis quantization (ACTCQ) method in [11]. PSNR versus rate points for competitive schemes, including the latter one, are also plotted in Fig. 3. The new results also surpass those in the original EZW [1] and are comparable to those for extended EZW in [6], which along with ACTCQ rely on arithmetic coding. The binary uncoded figures are only 0.3–0.6 dB lower in PSNR than the corresponding ones of the arithmetic coded versions, showing the efficiency of the partial ordering and set partitioning procedures. If one does not have access to the best CPU's and wishes to achieve the fastest execution, one could opt to omit arithmetic coding and suffer little consequence in PSNR degradation. Intermediary results can be obtained with, for example, Huffman entropy-coding. A recent work [12], which reports similar performance to our arithmetic coded ones at higher rates, uses arithmetic and trellis coded quantization (ACTCQ) with classification in wavelet subbands. However, at rates below about 0.5 bpp, ACTCQ is not as efficient and classification overhead is not insignificant.

Note in Fig. 3 that in both PSNR curves for the image LENA there is an almost imperceptible "dip" near 0.7 bpp. It occurs when a sorting pass begins, or equivalently, a new bit-plane begins to be coded, and is due to a discontinuity in the slope of the rate × distortion curve. In previous EZW versions [1], [6], this "dip" is much more pronounced, of up to 1 dB PSNR, meaning that their embedded files did not yield their best results for all rates. Fig. 3 shows that the new version does not present the same problem.

In Fig. 4, the original images are shown along with their corresponding reconstructions by our method (arithmetic coded only) at 0.5, 0.25, and 0.15 bpp. There are no objectionable artifacts, such as the blocking prevalent in JPEG-coded images, and even the lowest rate images show good visual quality. Table I shows the corresponding CPU times, excluding the time spent in the image transformation, for coding and decoding LENA. The pyramid transformation time was 0.2 s in an IBM RS/6000 workstation (model 590, which is particularly efficient for floating-point operations). The programs were not optimized to a commercial application level, and these times are shown just to give an indication of the method's speed. The ratio between the coding/decoding times of the different versions can change for other CPU's, with a larger speed advantage for the binary-uncoded version.

VIII. Summary and Conclusions

We have presented an algorithm that operates through set partitioning in hierarchical trees (SPIHT) and accomplishes completely embedded coding. This SPIHT algorithm uses the principles of partial ordering by magnitude, set partitioning by significance of magnitudes with respect to a sequence of octavely decreasing thresholds, ordered bit plane transmission, and self-similarity across scale in an image wavelet transform. The realization of these principles in matched coding and decoding algorithms is a new one and is shown to be more effective than in previous implementations of EZW coding. The image coding results in most cases surpass those reported previously on the same images, which use much more complex algorithms and do not possess the embedded coding property and precise rate control. The software and documentation, which are copyrighted and under patent applications, may be accessed in the Internet site with URL `http://ipl.rpi.edu/SPIHT` or obtained by anonymous ftp to `ipl.rpi.edu` with the path `pub/EW_Code` in the compressed archive file `codetree.tar.gz`. (The file must be decompressed with the command `gunzip` and exploded with the command "`tar xvf`"; the instructions are in the file `codetree.doc`.) We feel that the results of this coding algorithm with its embedded code and fast execution are so impressive that it is a serious candidate for standardization in future image compression systems.

References

[1] J. M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[2] M. Rabbani and P. W. Jones, *Digital Image Compression Techniques*. Bellingham, WA: SPIE Opt. Eng. Press, 1991.

[3] R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding," *IEEE Trans. Inform. Theory*, vol. 38, pp. 719–746, Mar. 1992.

[4] E. H. Adelson, E. Simoncelli, and R. Hingorani, "Orthogonal pyramid transforms for image coding," in *Proc. SPIE*, vol. 845, *Visual Commun. and Image Proc. II*, Cambridge, MA, Oct. 1987, pp. 50–58

[5] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, pp. 205–220, Apr. 1992.

[6] A. Said and W. A. Pearlman, "Image compression using the spatial-orientation tree," in *IEEE Int. Symp. Circuits and Systems*, Chicago, IL, May 1993, pp. 279–282.

[7] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–540, June 1987.

[8] P. Sriram and M. W. Marcellin, "Wavelet coding of images using trellis coded quantization," in *SPIE Conf. Visual Inform. Process.*, Orlando, FL, Apr. 1992, pp. 238–247; also "Image coding using wavelet transforms and entropy-constrained trellis quantization," *IEEE Trans. Image Processing*, vol. 4, pp. 725–733, June 1995.

[9] Y. H. Kim and J. W. Modestino, "Adaptive entropy coded subband coding of images," *IEEE Trans. Image Processing*, vol. 1, pp. 31–48, Jan. 1992.

[10] N. Tanabe and N. Farvardin, "Subband image coding using entropy-constrained quantization over noisy channels," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 926–943, June 1992.

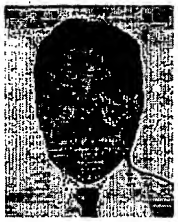
[11] R. L. Joshi, V. J. Crump, and T. R. Fischer, "Image subband coding using arithmetic and trellis coded quantization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 515–523, Dec. 1995.

[12] R. L. Joshi, T. R. Fischer, and R. H. Bamberger, "Optimum classification in subband coding of images," in *Proc. 1994 IEEE Int. Conf. Image Processing*, Austin, TX, Nov. 1994, vol. II, pp. 883–887.

[13] J. H. Kasner and M. W. Marcellin, "Adaptive wavelet coding of images," in *Proc. 1994 IEEE Conf. Image Processing*, Austin, TX, Nov. 1994, vol. 3, pp. 358–362.

[14] A. Said and W. A. Pearlman, "Reversible image compression via multiresolution representation and predictive coding," in *Proc. SPIE Conf. Visual Communications and Image Processing '93*, Cambridge, MA, Nov. 1993, Proc. SPIE 2094, pp. 664–674.

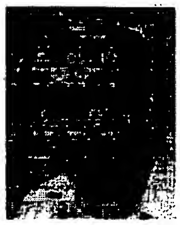
[15] D. P. de Garrido, W. A. Pearlman, and W. A. Finamore, "A clustering algorithm for entropy-constrained vector quantizer design with applications in coding image pyramids," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 83–95, Apr. 1995.



Amir Said (S'90–M'93) received the B.S. and M.S. degrees in electrical engineering from the State University of Campinas, Brazil, in 1985 and 1988, respectively. In 1994, he received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY.

After a year at IBM Brazil, in 1988 he joined the School of Electrical Engineering at the State University of Campinas, Brazil, as a Lecturer, and in 1994, he rejoined as an Assistant Professor. His research interests include image compression, channel coding and modulation, signal processing, and optimization.

Dr. Said was awarded the Allen B. DuMont prize for academic achievement from Rensselaer Polytechnic Institute.



William A. Pearlman (S'61–M'64–SM'84) received the S.B. and S.M. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, in 1963 and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1974.

Between 1963 and 1974 he was employed as an engineer and consultant by Lockheed Missiles and Space Company in Sunnyvale, CA, and as an engineer by GTE-Sylvania in Mountain View, CA. He left industry in 1974 to become an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Wisconsin-Madison. In 1979 he joined the Electrical, Computer, and Systems Engineering Department at Rensselaer Polytechnic Institute, where he is currently Professor. During the 1985–1986 academic year, and again for the spring semester in 1993, he was on sabbatical leave as Visiting Professor and Lady Davis Scholar in the Department of Electrical Engineering at the Technion-Israel Institute of Technology, Haifa, Israel. He also held a Visiting Professor Chair at Delft University of Technology in Delft, The Netherlands in January and February 1993 and has been an IBM Visiting Scientist in the IBM-Rio Scientific Center in Rio de Janeiro, Brazil in 1988. His research interests are in information theory, source coding theory, image, video, and audio coding, and image processing. He has authored or co-authored about 100 publications in these fields.

Dr. Pearlman has served on the steering committee of SPIE's Visual Communications and Image Processing Conference since its inception in 1986 and in 1989 was the conference chairman. he has also served on technical program committees of several IEEE conferences. He is currently Associate Editor of Coding for the IEEE TRANSACTIONS ON IMAGE PROCESSING. He is a Fellow of SPIE.

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore® RELEASE 1.5

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review Quick Links » Abstract Plus

Welcome to IEEE Xplore

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

Print Format

SEARCH RESULTS [PDF Full-Text (488 KB)] PREVIOUS NEXT
DOWNLOAD CITATION

A flexible zerotree coding with low entropy

Sang-Hyun Joo Kikuchi, H. Sasaki, S. Jaeho Shin
Dept. of Electr. Eng., Niigata Univ.;

This paper appears in: **Acoustics, Speech, and Signal Processing, 1998. ICASSP '98. Proceedings of the 1998 IEEE International Conference on**
Meeting Date: 05/12/1998 -05/15/1998
Publication Date: 12-15 May 1998
Location: Seattle, WA , USA
On page(s): 2685-2688 vol.5
Volume: 5, References Cited: 8
Number of Pages: 6 vol. lxiii+3816
INSPEC Accession Number: 6061785

Abstract:
We introduce a new zerotree scheme that effectively exploits the inter-scale self-similarities found in the octave decomposition by a wavelet transform. A zerotree is useful to code wavelet coefficients and its effectiveness was proved by Shapiro's (1993) EZW (embedded zerotree wavelet). In the coding scheme, wavelet coefficients are symbolized and then entropy-coded. The entropy per symbol is determined from the produced symbols and the final coded size is calculated by multiplying the entropy and the total number of symbols. We analyze symbols produced from the EZW and discuss the entropy per symbol. Since the entropy depends on the produced symbols, we modify the procedure of symbol generation. First, we extend the relation between a parent and children used in the EZW to raise the probability such that a significant parent has significant children. The proposed relation is flexibly extended according to the fact that a significant coefficient is likely to have significant coefficients in its neighborhood. Our coding results are compared with the published results of Shapiro and improvements come from the use of lower entropy per symbol. We also give a comparison of the number of produced symbols

Index Terms:
data compression entropy codes image coding transform coding wavelet transforms EZW children embedded zerotree wavelet entropy per symbol entropy-coded coefficients image coding inter-scale self-similarities low entropy octave decomposition parent probability significant coefficients symbol generation wavelet coefficients wavelet transform zerotree based compression zerotree coding

A FLEXIBLE ZEROTREE CODING WITH LOW ENTROPY

*Sang-hyun Joo, *Hisakazu Kikuchi, *Shigenobu Sasaki, **Jaeho Shin
* Department of Electrical Engineering, Faculty of Engineering, Niigata University, Japan
** Department of Electronics, Faculty of Engineering, Dongguk University, Korea

ABSTRACT We introduce a new zerotree scheme that effectively exploits the inter-scale self-similarities found in the octave decomposition by a wavelet transform. A zerotree is useful to code wavelet coefficients and its effectiveness was proved by Shapiro's EZW. In the coding scheme, wavelet coefficients are symbolized and then entropy-coded. The entropy per symbol is determined from the produced symbols and the final coded size is calculated by multiplying the entropy and the total number of symbols.

In this paper, we analyze symbols produced from the EZW and discuss the entropy per symbol. Since the entropy depends on the produced symbols, we modify the procedure of symbol generation. First, we extend the relation between a parent and children used in the EZW to raise the probability such that a significant parent has significant children. The proposed relation is flexibly extended according to the fact that a significant coefficient is likely to have significant coefficients in its neighborhood.

Our coding results are compared with the published results in paper [1] and improvements come from the use of lower entropy per symbol. We also give the comparison of the number of produced symbols.

KEYWORD: image compression, wavelet transform, zerotree coding

I. INTRODUCTION

In the wavelet-based coding, dependencies among bands using quadtrees were exploited in EZW(Embedded Zerotree Wavelet)[1], SPIHT(Set Partitioning In Hierarchical Trees)[2], SFQ(Space Frequency Quantization)[3,4]; that is, one coefficient at a given band is related with four coefficients at the same spatial location at the next finer band in terms of a relation between parent and children and the relation is applied for all coefficients except for DC coefficients. The EZW coder was designed by Shapiro who first applied an embedded zerotree using a wavelet. The algorithm is based on three concepts; 1) prediction of the absence of significant coefficients across scales by exploiting the self-similarity inherent in images 2) successive approximation for decoded coefficients 3) adaptive arithmetic coding for the streamed out symbols. After that, Said and Pearlman published their great work – SPIHT – that gives nice performances and fast processing. They use three lists to find significant coefficients in bands; 1) an LIP for insignificant coefficients 2) an LSP for significant coefficients 3) an LIS for insignificant descendants. The LIS includes two kinds of information for the descendants at the forms of type A or type B. The three lists are identically duplicated in a decoder by the transmitted bit stream. In the EZW and SPIHT algorithms, their merit is on the termination ability at any point that an encoder/decoder wants to stop, and the decoder reconstructs an approximated image from the information he has received. This property is clearly desirable when we consider of our constrained communication channels. More recently, Z.Xiong et. al. published an SFQ algorithm that surpasses the EZW and SPIHT in performance and there are two versions according to wavelet decompositions. One of them uses the octave band wavelet and the other uses the wavelet-packet. They get a coding performance while pruning branches from trees in a rate-distortion sense and scalar-quantizing the coefficients at the survived nodes. The decision to prune a branch or not depends on the pre-assigned bit budget and comparing of costs between pruning and non-pruning.

Most of coding schemes have two common procedures; 1) symbol generation (model transformation) and 2) entropy coding of the symbol stream. The symbol stream is produced for the purpose of representation and then symbols are entropy-coded. In this paper, we introduce a new zerotree scheme that lead lower entropy and thus more compression. Since the entropy per symbol is determined by the probabilities of produced symbols, we thus modify the procedure of the symbol generation with flexible treeing. The tree is flexibly designed in view of entropy. In the EZW scheme, a node on a tree branches out into four nodes and this relation is referred to as a fixed relation in the sense that the relation is not changed. On the other hand, our proposed relation is referred to as a "flexible tree" in the sense that a node on a tree branches into basic four nodes and flexibly extends its branches to nodes in neighbor. The idea to the flexible tree comes from how to extend more branches.

II. ZEROTREE BASED COMPRESSION

1. Embedded Zerotree Wavelet coding

Jerome M. Shapiro [1] developed an algorithm that exploits a relation between subbands in image compression. In the algorithm, zerotrees have been combined with bit plane coding and demonstrate the effectiveness of wavelet based coding. The algorithm is based on the zerotrees that efficiently represent many insignificant coefficients. As wavelet coefficients are located having some dependencies in bands, the dependencies are well exploited with a quadtree structure. The compression has three step procedures; 1) wavelet decomposition 2) symbol generation 3) entropy coding. We briefly review the coding algorithm and discuss produced symbols and its entropy. To describe the compression scheme, we quote several definitions - like parent, child, ancestor, descendant, root etc. - from the reference [1].

There are two types of passes performed: 1) a dominant pass 2) and a subordinate pass. The dominant pass finds significant coefficients to a given threshold, and the subordinate pass refines all significant coefficients found in all previous dominant passes. We use four symbols to tell a dominant pass to a decoder. A ZTR symbol is used for a zerotree root that is insignificant and has no significant descendants. One more needed symbol is an Isolated Zero symbol (named IZ) used when a coefficient is insignificant but has some significant descendants. Besides the symbols, two symbols are used for a significant coefficient – POS and NEG

0-7803-4428-6/98 $10.00 © 1998 IEEE

according to its sign. After all, the use of ZTR and IZ symbols is to inform locations of significant coefficients (POS and NEG) as efficiently as possible.

After a dominant pass, a subordinate pass is performed in order to refine the coefficients found to be significant in the previous dominant passes and these two passes are entropy-coded with an adaptive arithmetic coder[7].

2. The Shannon's entropy

As we reviewed in the previous section, a symbol stream is produced from the alternate passes and then the stream is entropy-coded for more compression. In this sub section, we briefly study the Shannon's entropy theorem to analyze the symbol stream.

Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of n symbols. Given $D = \{d_1, d_2, \ldots, d_l\}$, a data set of l symbols in a sequence (the number l is also called the data length of D), the probability distribution of the symbol set S in the data D is the collection of positive numbers $P = \{p_1, p_2, \ldots, p_n\}$, one for each symbol, defined by

$$p_i = |\{d_k \in D \mid d_k = s_i\}| / l, \quad for\ i = 1, 2, \cdots, n. \qquad (1)$$

If the probability distribution is the only assumed redundancy information, the pair (S,P) is called a zero-order Markov source. The data sequence D is called a zero-order Markov sequence.

Using the above notations, the (zero-order) entropy of the data sequence D is defined to be

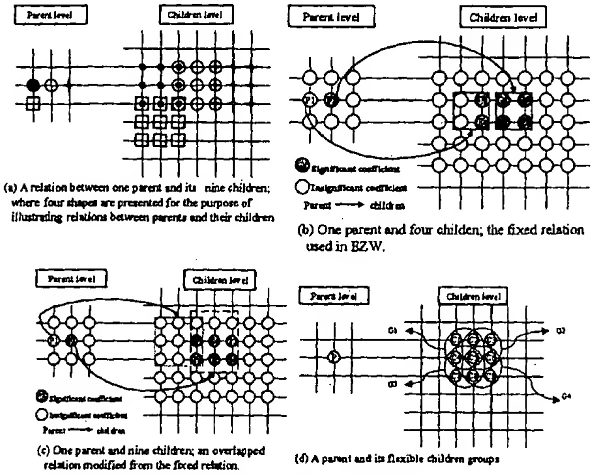
$$e(D) = -\Sigma p_i \cdot \log_2 p_i. \qquad (2)$$

3. A relation between the number of symbols and its entropy

With the Shannon's entropy, we consider a relation between the number of symbols and its entropy. The entropy per symbol largely depends upon occurrence probabilities of symbol alphabets and thus the final coded size is calculated by multiplying the average entropy and the number of symbols. Therefore, we can achieve more reduction in final coded size with two ways; one is to reduce the entropy per symbol and the other is to reduce the number of symbols.

We assume two particular source models for this discussion. Both models are composed of two symbols (S_1 and S_2) but the probability distributions and the numbers of symbols are different. Assume the first model has ten S_1 and ten S_2. In this case, the probabilities of symbols and its entropy are calculated by using equations 1 and 2. The model is assumed to be uniformly distributed and the entropy is 1 bit/symbol. Therefore, we should use 20 bits for the model. On the other hand, we assume the second model that has five S_1 symbols and 20-bit budget. In this case of the model, we consider how many S_2 symbols we can insert in the 20 bits. Using the equations 1 and 2, we can insert, at least, 20 S_2 symbols into the source. Therefore, the final output sizes are the same at 20 bits though they have different source length of symbols. It is important to compare the difference between the two distributions; in the second model, five S_1 has a worth of ten S_2 if we consider only the number of symbols. If the replacement is accomplished without any deformation of the contents, our attention will go to the number of symbol to be replaced. When we accomplish the replacement in smaller number than two, we do expect more compression with less entropy although the total number of symbols are increased. This consideration is discussed in the next section with more detailed example and we will apply to the EZW by using our flexible tree structure.

III. A FLEXIBLE RELATION IN PARENT-CHILD



(a) A relation between one parent and its nine children; where four shapes are presented for the purpose of illustrating relations between parents and their children

(b) One parent and four childen; the fixed relation used in EZW.

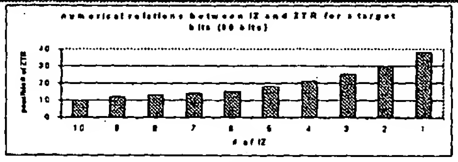
(c) One parent and nine children; an overlapped relation modified from the fixed relation.

(d) A parent and its flexible children groups

Figure III - 1. An example to explain a difference between the 1-4 relation and 1-9 relation.

Table III-1. The possible numbers of ZTR according to the variable numbers of IZ for a given budget (80 bits).

Cases	*# of POS*	*# of NEG*	*# of IZ*	*# of ZTR*	*Total # of symbols*	*Entropy (bits/sym.)*	*Coded size (bits)*
1	10	10	10	10	40	2.000	80
2	10	10	9	12	41	1.992	80
3	10	10	8	13	41	1.978	80
4	10	10	7	14	41	1.958	80
5	10	10	6	15	41	1.929	80
6	10	10	5	18	43	1.866	80
7	10	10	4	21	45	1.788	80
8	10	10	3	25	48	1.683	80
9	10	10	2	30	52	1.553	80
10	10	10	1	38	59	1.377	80



As reviewed in the previous section, a dominant pass in the EZW tells where significant coefficients with respect to a given threshold exist and which signs they have. In the pass, we use four symbols – ZTR, POS, NEG and IZ – to inform the locations and signs. Once an image is decomposed using a wavelet, the number of significant coefficients is decided. Therefore, it is the number of ZTR and IZ to decide length of a symbol stream and its entropy. We now consider the occurrence of these symbols. While a ZTR is produced when a coefficient and its descendants are insignificant, an IZ is produced when a coefficient is insignificant but some of descendants are significant.

Assumed that the numbers of POS and NEG are fixed as ten respectively, table III-1 shows relations in number between ZTR and IZ for a targeted budget (80 bits) in the views of numerical and graphical charts. As we can see in the table, when the numbers of IZ are decreased at a rate of one symbol, see how many ZTR can be coded into a stream. For a clear comparison, we give our attention to only the cases 1 and 10. In the case 1, the number of IZ are ten and therefore ten ZTR can be coded for the target size (80 bits). On the other hand, there is only 1 IZ and thus 38 ZTR can be coded for the same target size in the case 10. Although their target sizes are the same, we can code the different number of symbols. Comparing with the case 1, the case 10 can be interpreted as the decreased nine IZ symbols are replaced with the increased 28 ZTR symbols. The ratio 28 / 9 means that one IZ has the worth of 3.1 ZTR symbols. Therefore, we conclude a fact that it is more effective for an entropy coding to use three ZTR rather than one IZ, if and only if possible. Our interests are then on a possibility to such a symbol replacement and a ratio in the replacement.

To implement the symbol replacement, we have two step procedures; one is to decrease the number of IZ symbols and the other is to replace them with several ZTR symbols in order to compensate the decrease. We first suggest a solution to decrease IZ symbols. An insignificant coefficient is coded as an IZ when some descendants are significant. In other words, the occurrence of IZ is caused from one reason that the significant descendants belong to the insignificant ancestor. Therefore, a simple solution is to suppress the occurrence; let the significant descendants belong to a significant ancestor. It is only possible that the descendants have a power to select their ancestors. In the EZW, the relation does not allow such a selection and always maintains one parent to four children; this is referred as a fixed relation. The relation can be modified with another form so that some children can select their parent. Selecting a parent means that there should be several candidates for the parent and we can imagine that a modified relation must have an overlapped form. It can be made in various forms. One of them is suggested in figure III-1 (a). Four parents are displayed in the parent level having their own shapes. These shapes help to understand the relations between parents and their children. Each parent has nine children respectively and some children are shared by several candidates to be their parent; that is, a child can belong to one or more parents. In other words, we can scan a child after a parent among all candidates. This is referred as a modified relation; one parent–nine children.

We give a simple example to explain the modified relation. Assume that there are two parents at parent level – one is significant and the other is insignificant – and six significant children (named as C1 to C6) in child level as shown in the figure III-1 (b) and (c). Our goal is to find all significant coefficients according to the scanning order that we do not scan any children before any parents. We will use two relations to find them; the fixed and the modified relations. We have seven coefficients – one parent and six children – to be found as significant coefficients. We first find them with the fixed relation as shown in figure III-1 (b). The significant parent P2 has four significant children and they are scanned under P2. However, the insignificant parent P1 has also two significant children; thus the parent should be symbolized with an IZ symbol to find these two significant children. Therefore, we make a symbol stream in this case – IS (at parent level) SZSZ, SSSS (at child level); where S,I,Z mean a significant coefficient, an isolated zero and a zerotree root respectively. The stream has seven S, three Z and one I symbols. On the other hand, when the modified relation is applied to the example as shown in figure III-1 (c), all significant children belong to one significant parent and thus we need no IZ symbol. In this case, the symbol stream is output as ZS (at parent level) ZZZSSSSSS (at child level); seven S and four Z symbols. As was shown in the above explanation, two symbol streams were obtained for the same example by using two different relations. We knew that a relation between a parent and its children plays an important role in producing a symbol stream. According to specific relations, the kind and the number of produced symbols are different and thus the resulting entropy is different. In the cases of (b) and (c), entropies are 1.157 bit/symbol and 0.946 bit/symbol respectively. Their entropy coded sizes are 11.57 bits and 10.41 bits. Comparing those two streams, we conclude that one I symbol in the case (b) was replaced with two Z symbols in the case (c). After all, when we change a relation with another, an important thing is how many Z symbols are increased instead of decreasing I symbols. The ratio between the increase and decrease will be an important factor for an entropy coding.

To decrease the ratio, we again change the modified 1-9 relation with a flexible relation. In the previous example, the 1-9 relation was more efficient than the fixed relation as no use of I symbols. However, that is only the special example to explain a relation between I and Z symbols in numbers. If the P1 were also significant in the example, the symbol stream of the case (b) would not have included any I symbol and thus only two Z symbols are needed for the case. This means that the modified relation is not always better than the fixed relation is. Therefore, we need a general relation to compromise these two relations.

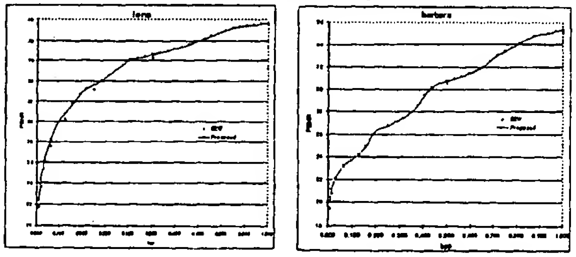
We exploit the dependencies in neighboring coefficients for that purpose. This can be realized by using a flexible relation; that is, the number of children a parent has is variable at a bound between four and nine. To define the flexible relation, we divide nine children into four groups that are named as G1,G2,G3 and G4 as shown in figure III-1 (d). A parent has the first group G1 and selectively has the rest groups of G2,G3 and G4; where each of rest groups – G2 ,G3 and G4 – is selected only when the first child in each group is significant. For example, G2 is selected when C2 is significant; in this case, the parent has G1 and G2 groups and six children C1 to C6 belong to the parent. Therefore, the G1 should be scanned before G2,G3 and G4. After all, selections of the rest groups G2,G3,G4 are determined by the significance of the children C2,C4,C5 in G1.

Back to the previous example, we apply the flexible relation. The first group G1 to the P1 has no significant children and thus P1 is coded as a zerotree root. The next parent P2 has two significant children C1 and C2 in G1; therefore, the children groups are G1,G3 and G4. In this case, the parent P2 has eight children except the second child of G2 among nine children. The resulting stream is ZS (at parent level) ZZSS (from G1) SS (from G3) SS (from G4); seven S and three Z symbols are included. Note that we do not need to scan a child twice. The flexible relation enables to decrease one more Z symbol than the modified relation.

IV. EXPERIMENTAL RESULTS

Our flexible tree is designed to reduce the number of IZ symbols and thus let the entropy lower. The decreased IZ symbols induce some increase of ZTR symbols in numbers by defining an extended relation. The ratio between the decrease and increase is efficiently exploited with the flexible treeing. We use two standard images – Lenna and Barbara (512 X 512 with a grey scaled level) – from the RPI site, ftp://ipl.rpi.edu/pub/image/still/usc. Our all results are based on 6-scaled octave wavelet transform and we use the 9/7 filter of [5] and mirror extensions at boundaries. Experimentally, the performances are compared with the published results at the reference [1] and they are plotted in fig IV-1 (a) and (b) for the two images respectively. The performances in PSNR are calculated over the ranges from 256 to 32768 Bytes. Our flexible coder shows 0.2~0.7 dB better performances than the EZW coder. The improvements are based on the symbol replacements by which we use a frequent symbol (ZTR) instead of infrequent symbol (IZ) as many as possible. We know the replacements are well accomplished with the flexible relation as shown in the performance curves.

In addition, we compare the number of symbols between the EZW and our coder. To give an exact comparison, we stop to code right after a threshold becomes 16; that is, the coding is terminated when the dominant and subordinate passes are all coded with respect to the threshold 32. The same condition is applied to the Barbara image and the results are given in table IV-1 (b). As we can see in the table, the numbers of POS and NEG symbols are the same but the compressed sizes are different. In the flexible relation, some IZ symbols are disappeared instead of some increase of ZTR symbols. As we can see in the table IV-1, we get different symbol streams from those two coders respectively. Comparing with the number of produced symbols in the EZW, our coder produce 1575 less IZ symbols and 4704 more ZTR symbols for the Barbara image. Therefore, the ratio can be calculated by dividing the increase in ZTR by the decrease in IZ; 4704 / 1575 = 2.99. The value 2.99 means that one IZ symbol was replaced with 2.99 ZTR symbols. The decreased IZ symbols play a part in lowering an entropy and therefore the image can be compressed with a smaller size. We can calculate each entropy for symbol distribution; 1.274 bits/sym. and 1.195 bits/sym. for the EZW and the proposed coder respectively. By using the low entropy, we can compress more compactly, though total number of symbols is increased.



(a) Lenna (b) Barbara

Figure IV-1. performance curves for the test images.

Table IV-1. Comparisons of produced symbols in numbers for the same number of significant coefficients.

Used coder	Compressed size (Bytes)	PSNR (dB)	Compression ratio	# of POS	# of NEG	# of ZTR	# of IZ
EZW	6511	32.51	40.26 : 1	3876	3743	43249	1566
Proposed	6328	32.51	41.43 : 1	3876	3743	46829	1369

(a) Lenna (512 X 512, 8 bits grey image, original size = 262144 Bytes)

Used coder	Compressed size (Bytes)	PSNR (dB)	Compression ratio	# of POS	# of NEG	# of ZTR	# of IZ
EZW	15878	30.05	16.51 : 1	9741	9586	72003	7603
Proposed	14584	30.05	17.97 : 1	9741	9586	76707	6028

(b) Barbara (512 X 512, 8 bits grey image, original size = 262144 Bytes)

V. CONCLUSIONS

We described a new relation that a parent takes its children with a flexible and selectable method. We extend the fixed relation used in the EZW scheme in order to decrease entropy per symbol. The ways to lower the entropy are accomplished by using more symbols that are frequent and less symbols that are infrequent. The infrequent symbol is IZ in the EZW and we can avoid the use of the symbol by extending the relation in parent-child; a parent has nine children and some of them are shared with neighboring parents. With the extended relation, the number of IZ symbol is decreased.

We showed that a symbol stream is coded with less entropy using the flexible relation in parent-child. Experimentally, our flexible coder has 0.2 ~ 0.7 dB better performances than the EZW's. We suppose that the flexible coder can be improved by a more efficient relation in parent-child.

ACKNOWLEDGMENTS: This work was in part supported by the Grant-in-Aid for Scientific Research, No. 07650419, from the Ministry of Education, Science and Culture of Japan and the research grant by Telecommunications Advancement Organization of Japan.

REFERENCES

1. J.M.Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, Vol. 41, No. 12, pp. 3445-3462, Dec. 1993
2. A.Said and W.A.Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video technology*, vol. 6, no. 3, pp. 243-250, June 1996.
3. Z.Xiong, K.Ramchandran and M.T.Orchard, "Space-Frequency Quantization for Wavelet Image Coding", *IEEE Trans. On Image Processing*, Vol. 6, No. 5, pp. 677-693, May 1997
4. Z.Xiong, K.Ramchandran and M.T.Orchard, "Wavelet packets image coding using space-frequency quantization" *IEEE Trans. on Image Processing*, January 1996.
5. I.H. Witten, R. Neal, and J.G. Cleary, "Arithmetic Coding for Data Compression", *Comm. ACM*, Vol. 30, No. 6, pp. 520 - 540, June 1987.
6. G. Strang, T. Nguyen, *Wavelets and filter banks*, Wellesley-Cambridge Press, 1996.
7. John W. Woods, *Subband image coding*, Kluwer Academic Publishers, Boston, MA, 1991.
8. S.H.Joo, H. Kikuchi, J.Shin, "A Flexible relationship in parent-child on embedded image coding", *Proceedings of IEICE Spring Conference*, vol. 1, no. 2, pp. 17, Mar. 1997.

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.5

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review Quick Links » Abstract Plus

Welcome to IEEE Xplore®

○ Home
○ What Can I Access?
○ Log-out

Tables of Contents

○ Journals & Magazines
○ Conference Proceedings
○ Standards

Search

○ By Author
○ Basic
○ Advanced

Member Services

○ Join IEEE
○ Establish IEEE Web Account
○ Access the IEEE Member Digital Library

Print Format

SEARCH RESULTS [PDF Full-Text (416 KB)] PREVIOUS NEXT
DOWNLOAD CITATION

Generalized bit-planes for embedded codes

da Silva, E.A.B. Craizer, M.
COPPE, Univ. Fed. do Rio de Janeiro;

This paper appears in: **Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on**
Meeting Date: 10/04/1998 -10/07/1998
Publication Date: 4-7 Oct 1998
Location: Chicago, IL , USA
On page(s): 317-321 vol.2
Volume: 2, References Cited: 8
Number of Pages: 3 vol. (lxxi+962+984+1013)
INSPEC Accession Number: 6201627

Abstract:

In many modern image compression techniques, the coefficients of an image transform are encoded by successive refinements. This is in general equivalent to successively encoding each bit-plane of the image transforms. Such methods, especially when the wavelet transform is employed, are among the state-of-the-art in image coding. In fact, several proposals to the JPEG 2000 standard use some sort of bit-plane encoding. Bit-plane encoding has been traditionally used as a scalar quantization technique, that is, each coefficient in individually decomposed in bit-planes. We analyze extensions of the bit-plane encoding concept to vectors, whereby each vector of the coefficients is decomposed in &ldquo;vector bit-planes&rdquo;. First, we formally describe the vector bit-plane representations, and then state a theorem concerning conditions for their existence. Next, we propose a modification to the proposed vector bit-plane representations and state a second theorem, which shows that the proposed modifications lead to much more robust algorithms. Simulation results are presented for the embedded encoding of wavelet coefficients of images, which confirm, the potential advantages of vector over scalar bit-plane representations. These strongly indicate that vector bit-plane representations should be further investigated

Index Terms:

data compression image coding image representation transform coding vector quantisation wavelet transforms JPEG 2000 standard VQ bit-plane encoding embedded codes embedded encoding generalized bit-planes image coding image compression image transform coefficients scalar quantization simulation results theorem

GENERALIZED BIT-PLANES FOR EMBEDDED CODES

Eduardo A. B. da Silva

PEE/COPPE/DEL/EE
Universidade Federal do Rio de Janeiro
Cx. P. 68504, Rio de Janeiro, RJ
21945-970, BRAZIL
eduardo@lps.ufrj.br

Marcos Craizer

Departamento de Matemática
PUC-Rio
R. Marquês de São Vicente, 225
Rio de Janeiro,RJ
22453-900, BRAZIL
craizer@saci.mat.puc-rio.br

ABSTRACT

In many modern image compression techniques, the coefficients of an image transform are encoded by successive refinements. This is in general equivalent to successively encoding each bit-plane of the image transforms. Such methods, especially when the wavelet transform is employed, are among the state-of-the-art in image coding. In fact, several proposals to the JPEG 2000 standard use some sort of bit-plane encoding. Bit-plane encoding has been traditionally used as a scalar quantization technique, that is, each coefficient in individually decomposed in bit-planes. In this paper we analyze extensions of the bit-plane encoding concept to vectors, whereby each vector of coefficients is decomposed in "vector bit-planes". First, we formally describe the vector bit-plane representations, and then state a theorem concerning conditions for their existence. Next, we propose a modification to the proposed vector bit-plane representations and state a second theorem, which shows that the proposed modifications lead to much more robust algorithms. Simulation results are presented for the embedded encoding of wavelet coefficients of images, which confirm the potential advantages of vector over scalar bit-plane representations. These strongly indicate that vector bit-plane representations should be further investigated.

1. INTRODUCTION

Wavelet transforms have been widely investigated for image coding applications. Among the wavelet image coding methods, the ones that are based on bit-plane encoding have become very popular. In these methods, the wavelet coefficients are coded in successive passes. In each pass, one bit-plane of the wavelet coefficients is encoded. In general, the similarities of the bands of same orientation is taken into consideration in the encoding of the bit-planes, by the use of zero-trees or similar structures. Good examples of bit-plane wavelet coders can be found in [1, 2, 3]. Although not restricted to wavelet coders, [4] is also a nice example of this recent trend to the use of bit-plane encoding. The performances of wavelet coders based on bit-plane encoding places them among the state-of-the-art in image coding.

Besides the good performance that can be obtained with bit-plane encoding of wavelet coefficients, bit-plane encoders have the advantage of naturally generating an embedded bitstream. In other words, an embedded bitstream with, for example, 1 bit/pixel, contains all the bitstreams with less than 1 bit/pixel. If we partially decode it up to 0.5 bit/pixel, we would obtain the same image that would be obtained if a bitstream of 0.5 bit/pixel was generated and decoded in the first place.

Bit-plane encoding is a form of scalar quantization, because each coefficient is individually decomposed into bit-planes. However, coding efficiency mandates that each bit-plane be encoded considering sets of coefficients. Zero-trees [1], run-length coding [3] and arithmetic encoding using conditional probabilities [5] are examples of that. Therefore, it is natural to wonder if there are non-trivial and efficient generalizations of bit-plane encoding to vectors. In other words, are there efficient ways of combining the advantages of bit-plane encoding and vector quantization? The answer to this question is affirmative, and in fact [6] describes a coder consisting of a straightforward substitution of the "scalar" bit-plane encoder in [1] by a kind of "vector" bit-plane encoder. The coder in [6] has shown an encouraging improvement in performance over the one in [1].

0-8186-8821-1/98 $10.00 © 1998 IEEE

In this paper, first the general problem of bit-plane encoding of vectors is analyzed. Then, two novel theorems which establish sufficient conditions for the vector bit-plane encoding be possible are stated, and some of its properties are analyzed. Based on these theorems, some improvements to the coder in [6], are proposed. Last, simulation results are presented, and extensions to this work are proposed.

2. VECTOR BIT-PLANES

To decompose a scalar quantity $-1 \leq c \leq 1$ using bit-planes is equivalent to represent it through a sequence $\{b_1, b_2, \ldots, b_n, \ldots\}$ such that:

$$c = s\sum_{i=1}^{\infty} b_i 2^{-i} \tag{1}$$

where $s \in \{-1, 1\}$ represents the sign of c and $b_i \in \{0, 1\}$.

Alternatively, we can have s always equal to 1 and $b_i \in \{1, -1\}$, yielding the representation below:

$$c = \sum_{i=1}^{\infty} b_i 2^{-i} \tag{2}$$

For example, in [2], $b_i \in \{0, 1\}$, and a representation like the one in eq. 1 is used. In [1], $b_i \in \{1, -1\}$, and coefficients are represented as in eq. 2.

In coding applications, the summation in eq. 2 is obviously not infinite, but goes from 1 to P, the number of bit planes, yielding the approximation c_P. In general, the more bit-planes are added to the summation, the smaller is the distortion $|c - c_P|$ in the representation of c. P is often chosen as the smallest value such that a certain distortion criterion is met, i.e. $|c - c_P| \leq \Delta$.

A trivial way to do vector bit-plane encoding of an N-dimensional vector $\mathbf{v}$ is to simply represent each of its coordinates v_k, for $k = 1, \ldots, N$ using bit-planes. Therefore, from eq. 2, we have that the vector $\mathbf{v}$ can be represented as:

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{\infty} b_{1i} 2^{-i} \\ \sum_{n=1}^{\infty} b_{2i} 2^{-i} \\ \vdots \\ \sum_{i=1}^{\infty} b_{Ni} 2^{-i} \end{pmatrix} \tag{3}$$

Eq. 3 is equivalent to:

$$\mathbf{v} = \sum_{i=0}^{\infty} \begin{pmatrix} b_{1i} \\ b_{2i} \\ \vdots \\ b_{Ni} \end{pmatrix} 2^{-i} = \sum_{i=0}^{\infty} \mathbf{b}_i 2^{-i} \tag{4}$$

We can see that eq. 4 is a vector version of eq. 2, that is, every vector $\mathbf{v}$ can be represented by a sequence of vectors $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n, \ldots\}$. If $b_{ki} \in \{1, -1\}$, $\mathbf{b}_i$ belongs to the codebook T_N whose vectors are of the form $\{((-1)^{\beta_1}\ (-1)^{\beta_2} \ldots (-1)^{\beta_N})^t\}$. Since all vectors b_i have magnitude equal to $\sqrt{N}$, they are located on an hyper-sphere, thus representing different orientations. For this reason, T_N can also be referred to as an *orientation codebook.*

Eq. 4 means that every vector $\mathbf{v}$ whose components are smaller than 1 can be represented as series of vectors of decreasing magnitudes (2^{-i}) and orientations drawn from a fixed orientation codebook T_N. As in the scalar case, the summations are not infinite, and go from 1 to the number of bit-planes P, yielding an approximation $\mathbf{v}_P$ of $\mathbf{v}$ as below:

$$\mathbf{v_P} = \sum_{i=0}^{Q} \mathbf{b}_i 2^{-i} \tag{5}$$

In coding applications, what is wanted is the smallest possible P (or, more exactly, the smallest possible entropy of the ensemble of vectors $\mathbf{b}_i$) such that the distortion $||\mathbf{v} - \mathbf{v}_P|| \leq \Delta$.

In this trivial extension of bit-plane encoding to vectors, the orientation codebook is composed by vectors whose components are 1 and -1 (eq. 4). At this point, a natural question to ask is whether there are other orientation codebooks such that vector bit-plane encoding is more efficient. More precisely, we are looking for representations of a vector $\mathbf{v}$ having the form:

$$\mathbf{v}_Q = \sum_{i=1}^{Q} \mathbf{u}_{n_i} \alpha^i \tag{6}$$

where $\mathbf{u}_{n_i} \in C_N$, an orientation codebook composed by unitary vectors on an hyper-sphere. We should also observe that the terms 2^{-i} in eq. 4 have been replaced by terms α^i, $0 < \alpha < 1$.

We want $\mathbf{v}_Q$ to arbitrarily approximate $\mathbf{v}$ for Q sufficiently large, that is,

$$\lim_{Q \to \infty} \mathbf{v}_Q = \mathbf{v} \tag{7}$$

Supposing that P and Q are the minimum number of terms in eqs. 5 and 6, respectively, such that the respective approximation errors are smaller than Δ. We have that the representation in eq. 6 is more efficient than the one in eq. 4 if the entropy of the ensemble of vectors $\mathbf{b}_i$, $i = 1, \ldots, P$ is larger than the one of the ensemble of vectors $\mathbf{u}_{n_i}$, $i = 1, \ldots, Q$, and vice versa.

In order to investigate more efficient vector bit-plane representations, one must first determine under what conditions does a general vector bit-plane representation as in eq. 6 exists such that eq. 7 holds. This will be dealt with in the next section.

3. EXISTENCE OF VECTOR BIT-PLANE REPRESENTATIONS

In order to determine conditions for the decompositions in eq. 6 to satisfy eq. 7 we have to first define one important parameter from an N-dimensional orientation codebook C_N, which we refer to as $\Theta(C_N)$[1]. It is the maximum possible angle between any vector $\in \mathbb{R}^N$ and its nearest neighbor $\in C_N$. More precisely,

$$\Theta(C_N) = \cos^{-1}\{\min_{\mathbf{x}\in\mathbb{R}^N}\{\max_{\mathbf{u}_i\in C_N}\{\frac{\mathbf{x}\cdot\mathbf{u}_i}{\|\mathbf{x}\|\|\mathbf{u}_i\|}\}\}\} \qquad (8)$$

We have then the following theorem:

Theorem 1 *Given an orientation codebook* $C_N = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M\}$ *such that* $\|\mathbf{u}_i\| = 1$, $\forall i$, *then there exists a representation as in eq. 6 such that eq. 7 is valid for all* $\mathbf{v} \in \mathbb{R}^N$, $\|\mathbf{v}\| \leq 1$ *if:*

$$\frac{1}{2\cos[\Theta(C_N)]} \leq \alpha < 1, \quad \Theta(C_N) \leq 45^\circ \qquad (9)$$

$$\sin[\Theta(C_N)] \leq \alpha < 1, \quad \Theta(C_N) \geq 45^\circ \qquad (10)$$

It is important to point out that this theorem only establishes sufficient conditions for eq. 7 to be valid for a representation as in eq. 6. In fact, its proof supposes worst case conditions. By worst case conditions it is meant that the angle between the residual $\mathbf{e}_i = \mathbf{v} - \mathbf{v}_i$ and $\mathbf{u}_{n_{i+1}}$ is supposed always to be equal to $\Theta(C_N)$, which is clearly very pessimistic.

An important property that can be deduced from eqs. 9 and 10 is that the smaller the value of $\Theta(C_N)$, the smaller α can be. We can estimate the impact of the values of α in eq. 6 by noting that, if eq. 7 is valid, the magnitude of the error of a Q-term approximation, $\mathbf{e}_Q = \mathbf{v} - \mathbf{v}_Q$ is given by

$$\|\mathbf{e}_Q\| = \|\sum_{i=Q+1}^{\infty} \mathbf{u}_{n_i}\alpha^i\| \leq |\sum_{i=Q+1}^{\infty} \alpha^i| = \frac{\alpha^{Q+1}}{1-\alpha} \qquad (11)$$

Eq. 11 shows us that the smaller the value of α, the smaller will be the bound on the residual approximation error. Therefore, from a coding point of view, it is interesting to have the smallest possible value of α in order to minimize the approximation error. Then, from eqs. 9 and 10 we have that the orientation codebook C_N should be such that value of $\Theta(C_N)$ is as small as possible. For any given dimension, there are two ways of reducing the values of $\Theta(C_N)$: (i) by increasing the number M of vectors; (ii) by distributing the vectors "more uniformly" over the N-dimensional unity sphere. However, when the number M of vectors in C_N is increased, there is a compromise: despite the decrease in the value of $\Theta(C_N)$, and, consequently, of α and the truncation error using Q vectors, there will be an increase in the entropy of the set $\mathbf{u}_{n_i}$, $i = 1, \dots, Q$. Therefore, for the entropy to be maintained Q would have to be reduced, thereby increasing the distortion. Thus, the best way to have a codebook with a low value of $\Theta(C_N)$ is to have its vectors the more "uniformly distributed" possible over the unity sphere in $\mathbb{R}^N$ for a given number M of vectors. Good examples of codebooks satisfying this property are the first shells of the lattices which solve the sphere packing problem in N dimensions [7]. For example, analyzing eq. 4, which describes the case of a mere concatenation of the bit-planes of the vector components, it can be proven that the codebook T_N has $\Theta(T_N) = \cos^{-1}(\sqrt{1/N})$. Table 1 compares the number of vectors and the values of $C(T_N)$ with the ones from the first shells of the lattices D_4, E_8 and Λ_{16}, which solve the sphere packing problem in dimensions 4, 8 and 16.

Codebook	T_4	D_4	T_8	E_8	T_{16}	Λ_{16}
M	16	24	256	240	65536	4320
Θ	60°	45°	69°	45°	76°	55°

Table 1: Values of Θ and number M of vectors for several orientation codebooks.

From this table we can see clearly the superiority of the codebooks derived from the lattices that solve the sphere packing problem. For example, T_8, despite having more vectors than E_8, has a much larger value of Θ. This implies that its vectors are much less uniformly distributed than the ones of E_8, and therefore vector bit-plane representations based on it are less efficient.

An algorithm for computing vector bit-plane representations

Theorem 1 described conditions for the existence of vector bit-plane representations as in eq. 6 satisfying eq. 7. An important point when it comes to practical applications is whether there is a fast algorithm for computing such representations. Fortunately, the answer is affirmative, and is given by the following greedy algorithm[2]:

[1] In [6] it has been referred to as $\theta_{\max}$.

[2] It should be noted that is is being assumed, without loss of generality, that $\|\mathbf{v}\| \leq 1$.

1. Make $m = 0$, $\mathbf{e}_0 = \mathbf{v}$ and $\beta = \alpha$.

2. Given the vector $\mathbf{e}_m$, choose $i_{m+1} \in \{1, \ldots, M\}$, where M is the size of the orientation codebook C_N, such that:

$$\mathbf{e}_m \cdot \mathbf{u}_{i_{m+1}} = \max\{\mathbf{e}_m \cdot \mathbf{u}_k : 1 \leq k \leq M\}$$

3. Compute $\mathbf{e}_{m+1} = \mathbf{e}_m - \beta \mathbf{u}_{i_{m+1}}$

4. Increment m, multiply β by α and go to step 2

One should note that this algorithm determines, in each pass, the vector $\mathbf{u}_{i_{m+1}}$ which is closest to the residual $\mathbf{e}_m$, and therefore minimizes the error in the representation of $\mathbf{e}_m$ in that pass. However, this procedure is not guaranteed to generate the optimum representation, that is, the one which yields the minimum representation error after Q passes. More precisely, if the algorithm above generates a sequence of vectors $u_{i_1}, u_{i_2}, \ldots, u_{iQ}$, which, according to eq. 6, provides an approximation $\mathbf{v}_{1Q}$, there is no guarantee that there is not a different sequence of vectors $u_{j_1}, u_{j_2}, \ldots, u_{jQ}$ providing an approximation $\mathbf{v}_{2Q}$ according to eq. 6 such that $||\mathbf{v} - \mathbf{v}_{2Q}|| < ||\mathbf{v} - \mathbf{v}_{1Q}||$. In other words, this discussion implies that, besides the fact that a representation as in eq. 6 satisfying eq. 7 is not unique, we have that the above algorithm will not necessarily find the optimum one. Fortunately, in most cases, the approximation it finds performs well enough.

At this point it is instructive to point out that the above algorithm has some similarities to Mallat's matching pursuit algorithm [8]. The main difference is that in Mallat's matching pursuits we replace the α^i term in eq. 6 by the projection of $\mathbf{e}_{i-1} = \mathbf{v} - \mathbf{v}_{i-1}$ on $\mathbf{u}_{n_i}$. More precisely, after Q passes, a matching pursuit decomposition of a vector $\mathbf{v}$ is of the form:

$$\mathbf{v}_Q = \sum_{i=1}^{Q} \gamma_i \mathbf{u}_{n_i} \qquad (12)$$

where, likewise eq. 6, $\mathbf{u}_{n_i} \in C_N$, an orientation codebook composed by unitary vectors on an hyper-sphere. On the other hand, unlike eq. 6,

$$\gamma_i = (\mathbf{v} - \mathbf{v}_{i-1}) \cdot \mathbf{u}_{n_i} \qquad (13)$$

This implies that, while in the vector bit-planes representation, a vector is represented by just a sequence of unity vectors $u_{i_1}, u_{i_2}, \ldots$, in Mallat's matching pursuits a vector is represented by a sequence of unity vectors $u_{j_1}, u_{j_2}, \ldots$ plus a sequence of projections $\gamma_{j_1}, \gamma_{j_2}, \ldots$

Performance of vector bit-plane encoding in the context of embedded wavelet coding

The vector bit-plane decomposition described above has been used in place of the conventional bit-plane decomposition in an EZW-like [1] algorithm. Details can be found in [6]. The results there have shown a performance improvement of around 1 dB for the LENA 512×512 image in the vicinity of 0.5 bit/pixel when the lattice Λ_{16} is used. As expected from the above discussions, the performance of the algorithm varies a great deal with α. For example, with Λ_{16}, the PSNR performance reaches a reasonably pronounced peak for α around 0.62 (see the "conventional algorithm" results in figure 1. This is in accord with what has been discussed above in relation with Theorem 1, because if α is too small eq. 7 is not valid, that is, the magnitude of the error $\mathbf{e}_Q$ does not tend to zero as $Q \to \infty$. If α gets too large, despite eq. 7 being valid, $||\mathbf{e}_Q||$ increases (see eq. 11). The main problem with this sort of behaviour is that the α value which gives peak performance is image dependent. That is inconvenient in a number of applications, especially when encoding time should be kept as low as possible.

In the next section we propose another theorem, together with a modification to the vector bit-plane decomposition which solves this problem, and provides almost peak performance for a large range of α values.

4. A MODIFIED VECTOR BIT-PLANE ENCODER

It can be shown that, in order for eq. 7 to hold for any $0 < \alpha < 1$, the algorithm above has to be modified in order to guarantee that the residual in pass m is such that[2] $\alpha^{m+1} \leq ||\mathbf{e}_m|| \leq \alpha^m$. This is done as follows: First, the zero vector is added to the codebook C_N. If the magnitude of $\mathbf{e}_m$ is smaller than α^{m+1}, $\mathbf{u}_{i_{m+1}}$ is chosen to be the zero vector, so that there is no refinement for that pass. If the magnitude of $\mathbf{e}_m$ is greater than α^m, β (see step 3) will not be multiplied by α in that pass. In a practical algorithm this can be signalized to the decoder by the inclusion of an escape code before the vector for that pass. With this modification, eq. 6 becomes:

$$\mathbf{v} = \sum_{i=1}^{\infty}\Big(\sum_{j=1}^{p(\mathbf{v},i)} \mathbf{u}_{n_{i_j}}\Big)\alpha^i \qquad (14)$$

Then, a new theorem can be stated, stronger then Theorem 1:

Theorem 2 *Suppose that the orientation codebook used in vector bit-plane encoding has $\Theta(C_N) \leq 60°$. Then* *a decomposition such as the one in eq. 14 exists for every* $0 < \alpha < 1$.

It is important to notice that, in order for the vector bit-plane decomposition in eq. 14 to be practical, $p(\mathbf{v}, i)$ should be small with great probability. For this reason, we used only $0.5 < \alpha < 1$ in our experiments. Indeed, we observed in these experiments that $p(\mathbf{v}, i) = 1$ occurred with a probability near 1 for these values of α.

Another point is that, likewise the algorithm derived from Theorem 1, this new algorithm does not necessarily lead to an optimal representation, but its performance is sufficiently good.

Experimental results

We have implemented the modifications leading to Theorem 2 to the vector bit-plane coder in [6]. The performance of this improved algorithm and the algorithm in [6] are compared in figure 1 for $0.5 \leq \alpha < 1$ using the first shell of the lattice Λ_{16} as the orientation codebook, for three different images. It can be observed that in the improved algorithm the performance does not degrade when α decreases, and therefore the choice of α is much less critical than in the previous case. In practice, this means that the value of α and can be made image independent, and therefore the modified algorithm is much more robust.

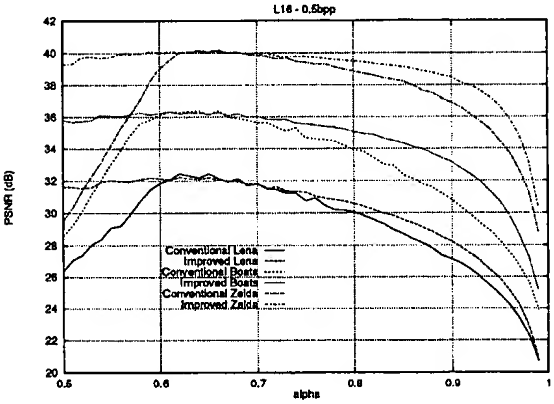


Figure 1: $\alpha \times PSNR$ for the images ZELDA, BOATS and LENA 256×256 at 0.5 bit/pixel with the "conventional" [6] and "improved"' (theorem 2) vector bit-plane coder, using the first shell of Λ_{16} as orientation codebook.

5. CONCLUSIONS

In this paper the concept of bit-plane encoding has been extended to vectors. It has been shown that there are orientation codebooks which can provide better performance than the trivial codebooks formed by separately bit-plane encoding each component of the vectors. Two theorems concerning the existence of "good" vector bit-plane decompositions have been stated. It has also been shown that when the first shells of the regular lattices which solve the sphere packing problem are used as orientation codebooks, the variations of vector bit-plane encoding proposed in this paper can outperform the conventional bit-plane encoding in EZW-like embedded wavelet encoders. However, the optimum choice of orientation codebooks still needs further investigation. Considering the great importance of bit-plane encoding in many modern image coding schemes, the results obtained are encouraging.

6. REFERENCES

[1] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 41, pp. 3445–3462, December 1993.

[2] A. Said and W. A. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.

[3] J. M. Tsai, J. Villasenor, and F. Chen, "Stack-run image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 519–521, October 1996.

[4] J. Andrew, "A simple and efficient hierarchical image coder," in *1997 IEEE International Conference on Image Processing*, (Santa Barbara, CA), October 1997.

[5] R. V. Algazi and R. E. Jr., "Analysis-based coding of image transforms and subband coefficients," in *Proceeding of the SPIE*, pp. 11–21, 1995.

[6] E. A. B. da Silva, D. G. Sampson, and M. Ghanbari, "A successive approximation vector quantizer for wavelet transform image coding," *IEEE Transactions on Image Processing, Special Issue on Vector Quantization*, vol. 5, pp. 299–310, February 1996.

[7] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. New York: Springer-Verlag, 1988.

[8] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397–3415, December 1993.

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.5

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review Quick Links » Abstract Plus

Welcome to IEEE Xplore®
○ Home
○ What Can I Access?
○ Log-out

Tables of Contents
○ Journals & Magazines
○ Conference Proceedings
○ Standards

Search
○ By Author
○ Basic
○ Advanced

Member Services
○ Join IEEE
○ Establish IEEE Web Account
○ Access the IEEE Member Digital Library

Print Format

SEARCH RESULTS [PDF Full-Text (672 KB)] PREVIOUS NEXT DOWNLOAD CITATION

Reduction of coding artifacts in transform image coding by using local statistics of transform coefficients

Choy, S.S.O. Yuk-Hee Chan
Dept. of Electron. Eng., Hong Kong Polytech. Univ., Kowloon;

This paper appears in: **Circuits and Systems, 1997. ISCAS '97., Proceedings of 1997 IEEE International Symposium on**
Meeting Date: 06/09/1997 -06/12/1997
Publication Date: 9-12 Jun 1997
, Hong Kong
On page(s): 1089-1092 vol.2
Volume: 2, References Cited: 7
Number of Pages: 4 vol. lxvi+2832
INSPEC Accession Number: 5738884

Abstract:
This paper proposes a new approach to reduce coding artifacts in transform image coding. We approach the problem in an estimation of each transform coefficient from the quantized data by using its local mean and variance. The proposed method can greatly reduce coding artifacts of low bit-rate coded images, and at the same time guarantee that the resulting image satisfies the quantization error constraint

Index Terms:
image coding quantisation (signal) transform coding coding artifacts local mean local statistics quantization error constraint quantized data transform coefficients transform image coding

Documents that cite this document
Select link to view other documents in the database that cite this one.

SEARCH RESULTS [PDF Full-Text (672 KB)] PREVIOUS NEXT DOWNLOAD CITATION

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms| Back to Top

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.5

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review Quick Links ▾ » Abstract Plus

Welcome to IEEE Xplore®

- ○ Home
- ○ What Can I Access?
- ○ Log-out

Tables of Contents

- ○ Journals & Magazines
- ○ Conference Proceedings
- ○ Standards

Search

- ○ By Author
- ○ Basic
- ○ Advanced

Member Services

- ○ Join IEEE
- ○ Establish IEEE Web Account
- ○ Access the IEEE Member Digital Library

Print Format

SEARCH RESULTS [PDF Full-Text (372 KB)] PREVIOUS NEXT
DOWNLOAD CITATION

An error resilient coding technique for JPEG2000

Hong Man Kossentini, F. Smith, M.J.T.
Dept. of ECE, Stevens Inst. of Technol., Hoboken, NJ;

This paper appears in: **Image Processing, 2000. Proceedings. 2000 International Conference on**
Meeting Date: 09/10/2000 -09/13/2000
Publication Date: 2000
Location: Vancouver, BC , Canada
On page(s): 364-367 vol.3
Volume: 3, References Cited: 4
Number of Pages: 3 vol.(lxviii+1027+957+1000)
INSPEC Accession Number: 7005371

Abstract:
We introduce a mixed fixed-length coding (FLC) and variable-length coding (VLC) technique to improve the error resilience of the upcoming JPEG2000 image coding standard. Our proposed method operates on quantized subband coefficients, and produces a bit-stream with both FLC and VLC codewords. Higher error resilience is achieved by eliminating the error propagation within FLC sections, which in general comprise the majority of the coded bit-stream. A highly efficient coding technique has been developed to generate fixed-length codewords for groups of quantization indices, and significantly reduce the amount of information to be entropy coded. However, the noiseless compression performance is not compromised. Every effect has been made to minimize the impact of our modifications to the baseline coding structure and the bit-stream syntax defined in the standard, so the proposed technique can be easily integrated into any JPEG2000 compliant system

Index Terms:
code standards data compression entropy codes image coding quantisation (signal) telecommunication standards transform coding variable length codes wavelet transforms JPEG2000 image coding standard baseline coding structure bit-stream bit-stream syntax dead-zone scalar quantization embedded block coding with optimized truncation entropy coding error propagation error resilient coding fixed-length coding noiseless compression performance quantization indices quantized subband coefficients variable-length coding wavelet/subband transformation

Documents that cite this document

Select link to view other documents in the database that cite this

AN ERROR RESILIENT CODING TECHNIQUE FOR JPEG2000 *

Hong Man[1] *Faouzi Kossentini* [2] *Mark J. T. Smith* [3]

[1]Department of ECE, Stevens Institute of Technology, Hoboken, NJ 07030
[2]Department of ECE, University of British Columbia, Vancouver BC V6T 1Z4, Canada
[3]School of ECE, Georgia Institute of Technology, Atlanta, GA 30332-0250

ABSTRACT

In this paper, we introduce a mixed fixed-length coding (FLC) and variable-length coding (VLC) technique to improve the error resilience of the upcoming JPEG2000 image coding standard. Our proposed method operates on quantized subband coefficients, and produces a bit-stream with both FLC and VLC codewords. Higher error resilience is achieved by eliminating the error propagation within FLC sections, which in general comprise the majority of the coded bit-stream. A highly efficient coding technique has been developed to generate fixed-length codewords for groups of quantization indices, and significantly reduce the amount of information to be entropy coded. However, the noiseless compression performance is not compromised. Every effect has been made to minimize the impact of our modifications to the baseline coding structure and the bit-stream syntax defined in the standard, so the proposed technique can be easily integrated into any JPEG2000 compliant system.

1. INTRODUCTION

The JPEG2000 coding algorithm consists of a wavelet/subband transformation, a dead-zone scalar quantization, and the "embedded block coding with optimized truncation" (EBCOT) [1] bit-plane entropy coding technique with a bit-stream truncation mechanism performing optimal rate control. The data streams flowing through these modules are lines of pixels for memory efficiency, which is refereed as the line-based processing. However in both transformation stage and entropy coding stage, temporary data pools are formed in order to perform the block-based operations.

After the subband decomposition and the scalar quantization, the subband coefficients are partitioned into blocks with default size of (64 × 64), and the coding of each block is independent to each other. The EBCOT algorithm employs a multi-pass bitplane entropy coding procedure. It operates on each individual blocks, and accesses each bitplane through three passes, namely the "Significance propagation pass (P_1)", the "Magnitude refinement pass (P_2)" and the "Normalization pass (P_3)". As a bitplane coding algorithm, a magnitude threshold is calculated for each bitplane, and it is scaled down by a factor of 2 at each successive bitplane. A pixel is identified as significant pixel at a certain bitplane if its magnitude is above the corresponding threshold, and a significant pixel will remain significant through all the rest of the bitplanes. At each bitplane, the P_1 passe only tests those insignificant coefficients which have at least one significant neighbor, and the P_3 pass tests all the rest of the insignificant coefficients. If a coefficient is found to become significant during either of these two passes, its sign bit will be encoded and its location can be inferred from all the test decision bits. The significant coefficients which are found in previous bitplanes are refined during the P_2 pass. All the symbols generated in these three passes will be encoded by an adaptive arithmetic coding engine with corresponding context modeling technique.

During preparation of the EBCOT encoding operation, every four rows of the quantized coefficients are interleaved into one row, and four consecutive insignificant coefficients will be encoded together to reduce the number of output symbols.

At the decoder, the EBCOT performs an inverse process in which coefficient values will be decoded and placed at their correct locations. These coefficients will then be de-interleaved to form the original subband blocks. This is followed by the inverse wavelet/subband transformation to obtain a reconstructed image.

2. PROPOSED CODING TECHNIQUE

Our work is based on the verification model VM5.2, and it is a special implementation of a generalized *Adaptive Quantization* scheme introduced in [2]. The proposed modifications preserve the line-based processing

*THIS WORK WAS SUPPORTED IN PART BY U.S. ARMY RESEARCH OFFICE UNDER GRANT # DAAH04-96-1-0161.

0-7803-6297-7/00/$10.00 © 2000 IEEE

pipeline, and the block-based EBCOT procedure. All the changes are strictly within the bitplane entropy coding stage.

During the coefficient interleaving process, we introduced a new operation, which perform a quantization index conversion after every four coefficients are read from a subband block. We do not impose any restriction on the order in which the block coefficients are scanned. The index conversion basically operates on every adjacent four coefficient group. Based on the current VM5.2 implementation, each of these four coefficient groups is essentially a vertical (4×1) vector in the original subband block. The quantization index conversion is based on the codebook of a specially designed 4-D multi-stage lattice vector quantizer (LVQ) [3].

An LVQ codebook contains a highly structured lattice codebook with points effectively spanning a particular signal space. It does not require any training in the design phase, and it can be implemented efficiently without codeword storage. By using a multistage quantization structure with relatively small codebook sizes, our LVQ does not yield any significant increase in computations complexity comparing to simple scalar quantizers.

Two LVQ codebooks have been designed for different quantization stages. Both are derived from the root lattice Z_4, which is the union of all integer points in the 4-dimensional space [4]. A 6-bit-per-vector sphere truncated LVQ is used for the first quantization stage which is able to achieve sufficient shape gain on generalized Gaussian source. The codebook is obtained by truncating the root lattice at the radius of 3, which produces an LVQ with 3 energy shells and 64 symbols. The energy measure for sphere truncation is the norm L_2. In order to include the origin in the codebook, one point is removed from the third shell. Thus, the total number of codewords is 64, requiring 6-bit indices for binary representation. A 4-bit-per-vector cubic LVQ is applied to all the successive refinement stages, which guarantees the convergence of all the successive approximations. Before truncation, the lattice is shifted by the vector $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$, so that all the corresponding quantizers become mid-rise SQs. The truncation radius is set to 1 in the infinity norm so that 16 codewords are obtained for each vector. As we can see that, by applying our LVQ to more than one stage, the resulting quantization cells are identical to those from the dead-zone scalar quantization, which confirms the compatibility of our LVQ in the baseline coding structure. The scaling factor between each consecutive quantization stages is $\frac{1}{2}$ in our multistage LVQ implementation, which essentially consists with the bit-plane coding in EBCOT.

In our modified EBCOT coding passes P_1 and P_3, only the first coefficient of each four coefficient group is tested and encoded. If this first coefficient is significant, all four coefficients are deemed as significant. Otherwise, all four coefficients remain as insignificant. The adaptive arithmetic coding of this significance test is exactly the same as the EBCOT approach. If the group is significant, we set the same context flags for all the four coefficients. The difference between our modified P_1, P_3 passes and the EBCOT passes is that we do not need to send sign bit of the new significant coefficients. This is because the sign information of each coefficient group is included in their first stage LVQ index. After the new significant coefficients are found at each bitplane, a first stage LVQ codeword is generated for each four coefficient group in the P_2 pass. Similar to the EBCOT, our P_2 pass also contains the refinement stage LVQ indices for all the previously found significant coefficient groups. It is important to point out that we do not use any arithmetic coding in the P_2 pass. This arrangement can be easily achieved because the baseline JPEG2000 defines a operation mood that uses only binary (raw) coding for the P_2 pass, therefore we do not have to introduce any new coding engine into the JPEG2000 coding structure. The final bit stream is organized in the way that the P_1 pass and P_3 passe represent all the arithmetic coded (VLC) sections, and the P_2 pass contains all the binary coded (FLC) section. Our modifications are transparent to the bit stream formation processes, and they do not effect any bit stream manipulation and other existing error resilience mechanism.

At the decoder, the P_1 and P_3 passes perform the arithmetic decoding of all the significance information for each four coefficient group at each bitplane. This information essentially identifies the locations of all the significant coefficients whose quantization indices have been included in the P_2 pass. It is apparent that once the significance of a coefficient group is found, the symbol length of its quantization index in the P_2 pass is known. Because of the binary coding used in the P_2 pass, there will be no confusion on which symbol belongs to which coefficient group. After all the decoding passes are completed, an inverse quantization index conversion is performed naturally during the corresponding de-interleaving process, and the original subband block is reconstructed.

3. ERROR RESILIENCE FEATURE

Effects of channel noise to source coded bit streams can be characterized into two categories, *propagational* and *non-propagational*. If a bit error can cause a loss of synchronization between the encoder and the decoder so that the bit stream following this error bit is entirely non-decodable, it is called propagational error. This usually happens when the source bit stream is coded through a variable length coding (VLC) scheme. On the other hand, a bit error in some bit stream may only cause a single error symbol at the decoder. This is called non-propagational error. Non-propagational error usually happens when fixed length coding (FLC) schemes are used. The problem with VLC in noisy channels is that it introduces inter-symbol dependencies within the coded bit stream. Correct decoding of each source symbol depends on the correctness of all the preceding decoded symbols. On the other hand, FLC schemes do not have this kind of problem. This is because the length of each source symbol is fixed and known to both encoder and decoder. Therefore every symbol can be independently decoded at the decoder. Any bit error in a FLC coded bit steam will be confined within a single symbol and will not effect the decoding of any other symbols. In general, the exclusive choice of either VLC or FLC represents a tradeoff between compression efficiency and error resilience. Our proposed scheme intends to explore a new approach in which a better balance between these two aspects can be achieved.

Based on our descriptions in the previous section, we can see that our modified P_1 and P_3 passes are VLC coded and the P_2 pass is FLC coded. In an error resilient mode, each pass is to be packetized, and there will be no error propagation cross the boundary of any two adjacent passes. Under this situation, the channel noise effects within our coded data stream will be quite different from those in the JPEG2000 bit stream. In the EBCOT coding passes, any bit error is in fact a propagational error because the extensive use of arithmetic coding. Therefore one bit error may cause the loss of a whole packet and all the rest of the packets in the same subband block. However in our bit stream, only bit errors in the P_1 and P_3 passes may cause similar damages. Bit errors in the P_2 pass are not propagational because of its FLC scheme, therefore they can only cause localized distortion, and no packet will be lost.

The advantage of this approach becomes apparent when our specially LVQ coding techniques can keep the VLC coded sections very small without any significant loss of compression performance. In our modified P_1 and P_3 passes, only one test symbol is generated for every four coefficients, and no sign bit is encoded. It is clear that the information processed by these passes is about 4+ times less than the amount encoded in the EBCOT approach. Therefore the resulting bit streams from P_1 and P_3 passes are much smaller than those produced by the EBCOT passes. During our extensive tests, we have found that for most images at commonly used bit rates, our coding scheme based on the four coefficient group with 4-D LVQ indexing usually produce about 20% ~ 30% VLC coded sections and 70% ~ 80% FLC coded sections, as shown in Tables 1 - 2. Comparing to the EBCOT approach where 100% of the bit stream is VLC coded, our data streams clearly have less chance of manifesting propagational errors. It is important to note that this error resilience feature does not rely on any error detection or correction scheme, and also will not reduce the effectiveness of any such scheme if they are employed.

	J2K VM5.2		Our Mod. VM5.2	
Image	PSNR(dB)	FLC %	PSNR(dB)	FLC %
WOMAN	27.27	0	26.81	76.39
CAFE	20.70	0	19.91	74.67
GOLD HILL	28.10	0	27.68	71.99

Table 1. PSNR performance of tested image coders at bit rate of 0.125 bpp in noiseless environment.

	J2K VM5.2		Our Mod. VM5.2	
Image	PSNR(dB)	FLC %	PSNR(dB)	FLC%
WOMAN	33.50	0	32.72	76.96
CAFE	26.69	0	25.67	73.91
GOLD HILL	32.78	0	31.87	75.53

Table 2. PSNR performance of tested image coders at bit rate of 0.5 bpp in noiseless environment.

4. EXPERIMENTAL RESULTS

We have tested the JPEG2000 VM5.2 coder and our modified VM5.2 coder in both noiseless and noisy channel situations. In the noisy channel simulations, a Binary Symmetric Channel (BSC) with random noise at BERs of 10^{-4} and 10^{-3} is used as the channel model. These BERs resemble some typical residual noise conditions in common error controlled communication networks. In both coder, the packet resynchronizations marker defined in the baseline is used as an error detection measure, and a simple error cancelation mechanism is used at the decoder which tends to discard any VLC packet with detected channel error. The images "WOMAN" (2048 × 2560), "CAFE" (2048 × 2560) and "GOLD HILL" (512 × 512) are coded at 0.125 bpp and 0.50 bpp. All noisy channel simulations are repeated 100 times for each test condition. Tables 1 -6 give the results of our simulations.

As we can see from these results, although our modified image coder produces a majority of FLC bit stream, it still performs reasonably well in noiseless channel comparing to the original VM5.2 coder. On

	PSNR (dB)	Mean	Max	Min	Std
J2K VM5.2	WOMAN	21.83	26.19	13.05	3.08
	CAFE	15.59	19.33	12.13	2.15
	GOLD HILL	26.67	28.10	17.89	1.39
Mod. VM5.2	WOMAN	23.39	26.55	13.20	2.74
	CAFE	17.66	19.74	12.69	1.77
	GOLD HILL	27.16	27.74	22.32	0.94

Table 3. PSNR performance of tested image coders at bit rate of 0.125 bpp in random noise BSC with $BER = 10^{-4}$.

	PSNR (dB)	Mean	Max	Min	Std
J2K VM5.2	WOMAN	21.36	26.39	13.07	2.92
	CAFE	15.44	20.76	11.77	2.15
	GOLD HILL	28.63	31.32	17.93	2.01
Mod. VM5.2	WOMAN	23.07	30.75	12.97	2.93
	CAFE	17.40	23.61	11.96	2.02
	GOLD HILL	29.60	31.85	20.87	2.73

Table 4. PSNR performance of tested image coders at bit rate of 0.5 bpp in random noise BSC with $BER = 10^{-4}$.

the other hand, we can clearly observe an performance improvement of up to 3 dB in various noisy channel.

Because our mixed FLC/VLC coding module operates on the same quantized subband coefficients as the VM coder does, and their rate control mechanisms are also the same, the difference between their noiseless performance is exclusively due to the different coding efficiencies between the FLC/VLC coding and the arithmetic coding. We notice that our LVQ coding method performs surprisingly well despite its fixed-length codeword and much simpler coding structure, comparing to sophisticate and high efficient arithmetic coding.

In general, our coder performs better at low bit rates and at low BERs. This is because at these situations, the VLC packets are very short, and therefore their surviving rates are high. On the other hand, when the bit rate and/or BER gets higher, even though our VLC packets are about 4 times shorter than their counterparts from the VM coder, they are still long enough to suffer from one or more bit errors, which consequently will cause the cancelation of the whole packet. This can be described as a noise saturated state for these VLC packets. It is easy to see that at a certain point when channel noise is so severe that almost all the packets will eventually be cancelled, and then our coder will have the same performance as the VM coder. To avoid this situation, we can apply forward error correction (FEC) to the bit-stream. With our short VLC packets and error resilient FLC packets, we can develop a more effective error protect scheme than what we can do with the JPEG2000 coder.

	PSNR (dB)	Mean	Max	Min	Std
J2K VM5.2	WOMAN	14.06	17.99	12.14	1.32
	CAFE	10.96	11.48	10.49	0.23
	GOLD HILL	21.77	25.11	16.43	2.19
Mod. VM5.2	WOMAN	17.28	22.19	12.19	3.26
	CAFE	12.55	15.18	10.51	1.27
	GOLD HILL	23.27	26.81	17.71	2.16

Table 5. PSNR performance of tested image coders at bit rate of 0.125 bpp in random noise BSC with $BER = 10^{-3}$.

	PSNR (dB)	Mean	Max	Min	Std
J2K VM5.2	WOMAN	13.87	17.95	12.14	1.41
	CAFE	10.97	11.48	10.58	0.26
	GOLD HILL	21.07	25.31	14.33	2.30
Mod. VM5.2	WOMAN	17.40	22.10	12.18	3.24
	CAFE	12.70	15.27	10.51	1.33
	GOLD HILL	22.51	26.85	14.80	2.33

Table 6. PSNR performance of tested image coders at bit rate of 0.5 bpp in random noise BSC with $BER = 10^{-3}$.

5. CONCLUSION

We have introduced a mixed FLC/VLC coding technique under the JPEG2000 image coding structure. It is able to achieve a comparable compression efficiency as the baseline coder has in noiseless environment, and yet provide better performance in noisy channels. Our modification is compatible to the basic coding structure and bit-stream syntax defined in the standard, and can be easily adopted into existing systems. Experimental results have clearly demonstrated the viability of our approach in both noiseless and noisy situations.

REFERENCES

[1] D. Taubman and A. Zakhor, "Multirate 3-d subband coding of video," *IEEE Trans. on Image Processing*, vol. 3, pp. 572–588, Sep. 1994.

[2] H. Man, M. Smith, and F. Kossentini, "On robustness of adaptive quantization for subband coding of images," in *SPIE Proc. Visual Communications and Image Processing*, (San Jose, CA), Jan. 1999.

[3] H. Man, F. Kossentini, and M. Smith, "A family of efficient and channel error resilient wavelet/subband image coders," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, pp. 95–108, Feb. 1999.

[4] J. D. Gibson and K. Sayood, "Lattice quantization," in *Advances in Electronics and Electron Physics, Vol 72* (P. W. Hawkes, ed.), pp. 259–330, Academic Press, Inc., 1988.

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

Help FAQ Terms IEEE Peer Review Quick Links » Abstract Plus

Welcome to IEEE Xplore®

○ Home
○ What Can I Access?
○ Log-out

Tables of Contents

○ Journals & Magazines
○ Conference Proceedings
○ Standards

Search

○ By Author
○ Basic
○ Advanced

Member Services

○ Join IEEE
○ Establish IEEE Web Account
○ Access the IEEE Member Digital Library

Print Format

SEARCH RESULTS [PDF Full-Text (60 KB)] PREVIOUS NEXT
DOWNLOAD CITATION

Global and local distortion inference during embedded zerotree wavelet decompression

Huber, A.K. Budge, S.E.
Dept. of Electr. & Comput. Eng., Utah State Univ., Logan, UT;

This paper appears in: **Data Compression Conference, 1996. DCC '96. Proceedings**
Meeting Date: 03/31/1996 -04/03/1996
Publication Date: Mar/Apr 1996
Location: Snowbird, UT , USA
On page(s): 441-
References Cited: 0
INSPEC Accession Number: 5335073

Abstract:
Summary form only given. Local distortion inference is proposed as an alternative to assuming a spatially uniform error for image compression applications requiring careful error analysis. This paper presents an algorithm for inferring estimates of the L_2-norm distortion (root-mean-square or RMS error) for the embedded zerotree wavelet (EZW) algorithm while maintaining its good rate-distortion performance. A small amount of both rate and computational burden is required of the encoder to calculate and transmit sums of wavelet coefficient energies. A greater computational burden is added to the decoder, mainly by an &ldquo;error-propagation transform&rdquo; of equal complexity to the inverse hierarchical wavelet transform. The asymmetry of the compression system with distortion inference is ideal for space-based data gathering applications where computation capacity may be limited in the encoder but virtually unlimited in the decoder. Global distortion inference is accomplished by maintaining error energy estimates of the wavelet coefficients separately for the subordinate and dominant lists. An equal reduction of error energy per significant bit is found to accurately interpolate the operational rate-distortion curve, which is explicitly transmitted prior to each dominant pass. Because of the orthogonality of the wavelet transform, the rate-distortion curve also applies to reconstructed images in the spatial domain

Index Terms:
coding errors data compression decoding error analysis estimation theory image coding image reconstruction inference mechanisms rate

Global and Local Distortion Inference During Embedded Zerotree Wavelet Decompression

A. Kris Huber and Scott E. Budge
Electrical and Computer Engineering Dept.
Utah State University
Logan, Utah 84322–4120
e-mail: kris@ece.usu.edu & scott@goga.ece.usu.edu

Abstract

Local distortion inference is proposed as an alternative to assuming a spatially uniform error for image compression applications requiring careful error analysis. This paper presents an algorithm for inferring estimates of the L_2-norm distortion (root-mean-square or RMS error) for the Embedded Zerotree Wavelet (EZW) algorithm while maintaining its good rate-distortion performance. A small amount of both rate and computational burden is required of the encoder to calculate and transmit sums of wavelet coefficient energies. A greater computational burden is added to the decoder, mainly by an "error-propagation transform" of equal complexity to the inverse hierarchical wavelet transform. The asymmetry of the compression system with distortion inference is ideal for space-based data gathering applications where computation capacity may be limited in the encoder but virtually unlimited in the decoder.

Global distortion inference is accomplished by maintaining error energy estimates of the wavelet coefficients separately for the subordinate and dominant lists. An equal reduction of error energy per significant bit is found to accurately interpolate the operational rate-distortion curve, which is explicitly transmitted prior to each dominant pass. Because of the orthogonality of the wavelet transform, the rate-distortion curve also applies to reconstructed images in the spatial domain. No additional rate overhead is needed to obtain a local distortion estimate. Individual estimates of wavelet coefficient error energies may be transformed to the spatial domain by applying the statistical propagation of errors formula for weighted sums of random variables. The resulting local distortion information is a "noise image" which gives an estimate of the RMS error for each pixel of the decompressed image. This local information is analogous to the error-bars often plotted on graphs of 1-dimensional data. It can be used during an analysis to more appropriately weight the value of each pixel, rather than weighting them all equally.

This local distortion estimate is most useful at low bit rates, when compression error dominates. Large errors, however, can occur that are significantly underestimated by the noise image. This is probably caused by correlation of the error with the input image and/or nearby errors. The error propagation transform, as implemented, assumes errors are uncorrelated since no correlation information is available at the decoder. In spite of this defect, the local estimate is usually a better estimate than the global RMS error for low bit rates, even for the outliers. At higher bit rates the EZW compression error becomes very Gaussian and quite spatially uniform. In this case the local distortion estimate gives little or no improvement over the global estimate.

1068-0314/96$5.00 © 1996 IEEE

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.5

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review Quick Links » Abstract Plus

Welcome to IEEE Xplore®

- ○ Home
- ○ What Can I Access?
- ○ Log-out

Tables of Contents

- ○ Journals & Magazines
- ○ Conference Proceedings
- ○ Standards

Search

- ○ By Author
- ○ Basic
- ○ Advanced

Member Services

- ○ Join IEEE
- ○ Establish IEEE Web Account
- ○ Access the IEEE Member Digital Library

Print Format

SEARCH RESULTS [PDF Full-Text (400 KB)] PREVIOUS NEXT
DOWNLOAD CITATION

On the error concealment technique for DCT based image coding

Jong Wook Park Dong Sik Kim Sang Uk Lee
Signal Process. Lab., Seoul Nat. Univ.;

This paper appears in: **Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on**
Meeting Date: 04/19/1994 -04/22/1994
Publication Date: 19-22 Apr 1994
Location: Adelaide, SA , Australia
On page(s): III/293-III/296 vol.3
Volume: iii, References Cited: 7
INSPEC Accession Number: 4929218

Abstract:

This paper presents a new error concealment (EC) technique for DCT based image coding. In our approach, the damaged blocks are recovered utilizing the smoothness property of an image at the boundaries of the blocks. Based on the property, we first define an object function which represents the intersample variations between adjacent blocks. Then, the DCT coefficients which minimize the object function are estimated by finding a solution of a linear equation. And, we show that it can be decomposed and reduced into more simple sub-equations. Thus, the computational complexity of the proposed technique is very low, compared to the existing techniques. Computer simulation results show that the proposed algorithm recovers the damaged blocks even if the block loss rate (BLR) is as high as 10^{-2}

Index Terms:

DCT based image coding block codes block loss rate computational complexity computer simulation damaged blocks discrete cosine transforms error concealment technique error correction codes image coding intersample variations object function smoothness property transform coding visual perception DCT based image coding block codes block loss rate computational complexity computer simulation damaged blocks discrete cosine transforms error concealment technique error correction codes image coding intersample variations object function smoothness property transform coding visual perception

Documents that cite this document

Select link to view other documents in the database that cite this one.

ON THE ERROR CONCEALMENT TECHNIQUE FOR DCT BASED IMAGE CODING

Jong Wook Park *Dong Sik Kim* *Sang Uk Lee*
Signal Processing Lab. Dept. of Cont. & Inst. Eng.
Seoul National University, Seoul 151-742, KOREA

ABSTRACT

This paper presents a new error concealment (EC) technique for DCT based image coding. In our approach, the damaged blocks are recovered utilising the smoothness property of an image at the boundaries of the blocks. Based on the property, we first define an object function which represents the intersample variations between adjacent blocks. Then, the DCT coefficients which minimise the object function are estimated by finding a solution of a linear equation. And, we show that it can be decomposed and reduced into more simple sub-equations. Thus, the computational complexity of the proposed technique is very low, compared to the existing techniques. Computer simulation results show that the proposed algorithm recovers the damaged blocks, and there exists no longer annoying in visual perception, even if the block loss rate (BLR) is as high as 10^{-2}.

1. INTRODUCTION

Most of recent image coding standards are based on the discrete cosine transform (DCT) and variable length coding (VLC) to compress an image data. In transmitting the compressed image data, even 1 bit error in the bit-stream may cause a considerable damage in the reconstructed image qualities. Thus, a scheme to alleviate the effect of inevitably occurring errors are necessary for a reliable image communication. In general, there are two approaches: the forward error correction (FEC), and EC techniques. The EC attempts to fill the damaged areas with its estimated one, which is usually obtained from the spatially or temporally neighboring data. Thus, the EC can recover the errors uncorrected by FEC, and does not require any additional information.

Several EC techniques have been proposed [1-4], which are based on the temporal replacement (TR) or spatial interpolation (SI) techniques. The TR technique is known to be very effective in slow-motion area, while the SI technique is more preferred to the fast-moving objects. In SI techniques, however, it is of important to reconstruct the plausible DCT coefficients of damaged blocks. Sun [2] adopted the blocking effect reduction algorithm of JPEG [7]. Wang et al. introduced the optimisation technique based on the smoothness property of an image [1], and Lee et al. proposed to the use of a fussy logic reasoning [4].

In this paper, we present a new EC technique for DCT based image coding. In our approach, we first define an object function to describe the intersample variation at the boundary between adjacent blocks. Then, by estimating the DCT coefficients which minimise the object function in least square (LS) sense, the recovered block is smoothly connected to the neighboring blocks. The computer simulation results indicate that the recovered blocks are not easily recognised, even the block loss rate (BLR) is as high as order of 10^{-2}, where the BLR is the ratio of average number of damaged blocks to the number of total received blocks [6]. Moreover, the computational complexity of the proposed algorithm is an order of $O(N)$, where N is the block sise, making a real-time implementation possible.

2. OBJECT FUNCTION FOR DCT COEFFICIENT RECOVERY

The effects of bit errors in bit-stream are very complex and unpredictable. Thus, in order to simplify the problem, we introduce several assumptions. First, we employ the well-known smoothness assumption on the luminance level within a small area of image, as in [1]. The second assumption is that all the DCT coefficients in the erroneous block are completely lost. The third one is that the locations of erroneous blocks can be isolated, provided that a proper block interleaving algorithm is used, such as [5]. Last, the locations of damaged blocks are known *a priori*, since it is possible to check whether the block is erroneous or not by examining the protocol, synchronising codeword, and etc.

Next, let us introduce some notations to represent the blocks and vectors for the sake of convenience. Let the sise of a block be $N \times N$, and an image is composed of $Q \times P$ array of blocks. By denoting the block at time τ by A_τ, then the upper, lower, left, and right block of A_τ are denoted by $A_{\tau-P}$, $A_{\tau+P}$, $A_{\tau-1}$, and $A_{\tau+1}$, respectively. Also let the 4 boundary vectors of A_τ in 4 different directions, namely, top, bottom, left, and right be t_τ, b_τ, l_τ, and r_τ, respectively. All the defined blocks and vectors are described in Fig. 1.

Now, we define an object function for the estimation of the lost DCT coefficients. Based on the smoothness assumption, it is believed that the recovered block should be smoothly connected to the neighboring blocks at the boundaries. Hence, an object function is defined as

$$\begin{aligned}\psi = & \parallel t_\tau - b_{\tau-P} \parallel^2 + \parallel b_\tau - t_{\tau+P} \parallel^2 \\ + & \parallel l_\tau - r_{\tau-1} \parallel^2 + \parallel r_\tau - l_{\tau+1} \parallel^2 . \end{aligned} \qquad (1)$$

Notice that the object function measures the degree of the smoothness. From the energy preserving property of

0-7803-1775-0/94 $3.00 © 1994 IEEE

the DCT, the object function can be rewritten in the DCT domain as

$$\begin{aligned}\Psi &= \| T_\tau - B_{\tau-P} \|^2 + \| B_\tau - T_{\tau+P} \|^2 \\ &+ \| L_\tau - R_{\tau-1} \|^2 + \| R_\tau - L_{\tau+1} \|^2, \qquad (2)\end{aligned}$$

where T_τ, B_τ, L_τ, and R_τ are 1-dimensional DCT pairs of the boundary vectors t_τ, b_τ, l_τ, and r_τ, respectively. On the other hand, after some algebraic manipulations and from the definition of DCT, we obtain the following relations:

$$\begin{aligned}T_\tau(j) &= \sum_{k=0}^{N-1} \beta_k \mathbf{A}_\tau(k,j), \quad j = 0,\cdots,N-1. \qquad (3a)\\ B_\tau(j) &= \sum_{k=0}^{N-1} s_k \beta_k \mathbf{A}_\tau(k,j), \quad j = 0,\cdots,N-1, \qquad (3b)\\ L_\tau(i) &= \sum_{l=0}^{N-1} \beta_l \mathbf{A}_\tau(i,l), \quad i = 0,\cdots,N-1, \qquad (3c)\\ R_\tau(i) &= \sum_{l=0}^{N-1} s_l \beta_l \mathbf{A}_\tau(i,l), \quad i = 0,\cdots,N-1, \qquad (3d)\end{aligned}$$

where

$$\beta_k = \alpha(k) \cos \frac{k\pi}{2N}, \qquad s_k = \begin{cases} 1, & \mathrm{k = even}, \\ -1, & \mathrm{k = odd}, \end{cases} \qquad (4)$$

and $\alpha(0) = \sqrt{\frac{1}{N}}$, and $\alpha(\cdot) = \sqrt{\frac{2}{N}}$, otherwise. Also $\mathbf{A}_\tau$ denotes 2-dimensional DCT pairs of A_τ.

By substituting (3) into (2) and applying the l_2 norm, the object function becomes

$$\begin{aligned}\Psi &= \sum_{j=0}^{N-1}[\sum_{k=0}^{N-1} \beta_k \mathbf{A}_\tau(k,j) - \sum_{k=0}^{N-1} s_k \beta_k \mathbf{A}_{\tau-P}(k,j)]^2 \\ &+ \sum_{j=0}^{N-1}[\sum_{k=0}^{N-1} s_k \beta_k \mathbf{A}_\tau(k,j) - \sum_{k=0}^{N-1} \beta_k \mathbf{A}_{\tau+P}(k,j)]^2 \\ &+ \sum_{i=0}^{N-1}[\sum_{l=0}^{N-1} \beta_l \mathbf{A}_\tau(i,l) - \sum_{l=0}^{N-1} s_l \beta_l \mathbf{A}_{\tau-1}(i,l)]^2 \\ &+ \sum_{i=0}^{N-1}[\sum_{l=0}^{N-1} s_l \beta_l \mathbf{A}_\tau(i,l) - \sum_{l=0}^{N-1} \beta_l \mathbf{A}_{\tau+1}(i,l)]^2. \qquad (5)\end{aligned}$$

But, it is worth to note that (5) is a function of N^2 variables in $\{\mathbf{A}_\tau(k,l) | k,l = 0,\cdots,N-1\}$.

3. ESTIMATION OF THE DCT COEFFICIENT

In this section, we shall present an algorithm to calculate $\mathbf{A}_\tau$ which minimizes the object function Ψ. By examining (5), we can see that Ψ is a quadratic function of each $\mathbf{A}_\tau(m,n)$, where $m,n = 0,\cdots,N-1$. Therefore, we can find the minimum value of Ψ, where all the gradients vanish. By partial differentiating both sides of (5) with respect to $\mathbf{A}_\tau(m,n)$, we can obtain the following N^2 equations:

$$\begin{aligned}&2(\beta_m^2 + \beta_n^2)\mathbf{A}_\tau(m,n) \\ &+ \sum_{k=0,k\neq m}^{N-1} \beta_m \beta_k (1 + s_m s_k) \mathbf{A}_\tau(k,n) \\ &+ \sum_{l=0,l\neq n}^{N-1} \beta_n \beta_l (1 + s_n s_l) \mathbf{A}_\tau(m,l) \\ &= \phi(m,n), \qquad m,n = 0,\cdots,N-1, \qquad (6)\end{aligned}$$

where

$$\begin{aligned}\phi(m,n) &= \sum_{k=0}^{N-1} \beta_m \beta_k [s_k \mathbf{A}_{\tau-P}(k,n) + s_m \mathbf{A}_{\tau+P}(k,n)] \\ &+ \sum_{l=0}^{N-1} \beta_n \beta_l [s_l \mathbf{A}_{\tau-1}(m,l) + s_n \mathbf{A}_{\tau+1}(m,l)]. \qquad (7)\end{aligned}$$

However, all the N^2 linear equations in (6) can be represented by a matrix-vector form, given by

$$\mathbf{Hf} = \mathbf{\Phi}, \qquad (8)$$

where $\mathbf{H}$ is a matrix of $N^2 \times N^2$, each row of which represents one of the linear equations in (6). And $\mathbf{f}$ is a vector composed of the coefficients of $\{\mathbf{A}_\tau\}$, and similarly, $\mathbf{\Phi}$ is a vector composed of $\{\phi\}$. Hence, by solving the linear equation (8), we can estimate the lost coefficients $\mathbf{A}_\tau$, which minimize the Ψ in the LS sense.

In order to solve (8), the inversion of $\mathbf{H}$ is necessary. But, in most case the size of $\mathbf{H}$ is so large that the inversion is difficult and unreliable. Therefore, it is necessary to obtain the solution indirectly by making $\mathbf{H}$ more simple form by following two steps. First, let us decompose the matrix $\mathbf{H}$ into 4 sub-matrices. Among the coefficients of $\{\mathbf{A}_\tau(k,n_1) | k = 0,\cdots,N-1\}$ involved in the computation of $\mathbf{A}_\tau(m_1,n_1)$ for given constant integer m_1, n_1, only the coefficients k of which satisfies the following condition are selected.

$$k \;\; mod \;\; 2 \;\; = \;\; m_1 \;\; mod \;\; 2. \qquad (9)$$

Note that depending on the value of k, the term $(1+s_{m_1}s_k)$ in (6) becomes zero if the condition (9) is not satisfied. Similarly, the condition for l is given by

$$l \;\; mod \;\; 2 \;\; = \;\; n_1 \;\; mod \;\; 2, \qquad (10)$$

which can be derived from the term $(1 + s_{n_1} s_l)$ in (6). It can be shown that these two conditions make it possible to group the elements of $\{\mathbf{A}_\tau\}$ into 4 classes, namely,

$$\begin{aligned}S_1 &= \{A_\tau(m,n) | \;\; m = even, \;\; n = even\}, \\ S_2 &= \{A_\tau(m,n) | \;\; m = even, \;\; n = odd\}, \\ S_3 &= \{A_\tau(m,n) | \;\; m = odd, \;\; n = even\}, \\ S_4 &= \{A_\tau(m,n) | \;\; m = odd, \;\; n = odd\}. \qquad (11)\end{aligned}$$

Then, it is easy to show that the matrix $\mathbf{H}$ can be decomposed into 4 sub-matrices. As a results, (8) is divided into 4 sub-equations, *i.e.*,

$$\mathbf{H}_i \mathbf{f}_i \;\; = \;\; \mathbf{\Phi}_i\,, \qquad i = 1,2,3,4, \qquad (12)$$

where f_i is composed of the elements of S_i, Φ_i is a vector composed of the elements of Φ corresponding to f_i, respectively.

But, each sub-matrix can be further simplified by ignoring the high frequency DCT coefficients, since dominant low frequency DCT coefficients are sufficient to reconstruct the damaged block within tolerable degradation. Thus, if we calculate only ν coefficients from each of the 4 sub-equations, then the dimension of each sub-matrix is reduced to $\nu \times \nu$, resulting in a considerable saving in computational complexity.

For example, in the case of $\nu = 3$, only 12 coefficients shown in Fig.2 are calculated. In this case, 4 sub-equations are given by,

$$\begin{bmatrix} 4\beta_0^2 & 2\beta_0\beta_2 & 2\beta_0\beta_2 \\ 2\beta_2\beta_0 & 2(\beta_0^2+\beta_2^2) & 0 \\ 2\beta_2\beta_0 & 0 & 2(\beta_2^2+\beta_0^2) \end{bmatrix} \begin{bmatrix} A_{00} \\ A_{02} \\ A_{20} \end{bmatrix} = \begin{bmatrix} \phi_{00} \\ \phi_{02} \\ \phi_{20} \end{bmatrix}. \quad (13a)$$

$$\begin{bmatrix} 2(\beta_0^2+\beta_1^2) & 2\beta_1\beta_3 & 2\beta_0\beta_2 \\ 2\beta_3\beta_1 & 2(\beta_0^2+\beta_3^2) & 0 \\ 2\beta_2\beta_0 & 0 & 2(\beta_1^2+\beta_2^2) \end{bmatrix} \begin{bmatrix} A_{01} \\ A_{03} \\ A_{21} \end{bmatrix} = \begin{bmatrix} \phi_{01} \\ \phi_{03} \\ \phi_{21} \end{bmatrix}, \quad (13b)$$

$$\begin{bmatrix} 2(\beta_1^2+\beta_0^2) & 2\beta_0\beta_2 & 2\beta_1\beta_3 \\ 2\beta_2\beta_0 & 2(\beta_1^2+\beta_2^2) & 0 \\ 2\beta_3\beta_1 & 0 & 2(\beta_3^2+\beta_0^2) \end{bmatrix} \begin{bmatrix} A_{10} \\ A_{12} \\ A_{30} \end{bmatrix} = \begin{bmatrix} \phi_{10} \\ \phi_{12} \\ \phi_{30} \end{bmatrix}, \quad (13c)$$

$$\begin{bmatrix} 4\beta_1^2 & 2\beta_1\beta_3 & 2\beta_1\beta_3 \\ 2\beta_3\beta_1 & 2(\beta_1^2+\beta_3^2) & 0 \\ 2\beta_3\beta_1 & 0 & 2(\beta_3^2+\beta_1^2) \end{bmatrix} \begin{bmatrix} A_{11} \\ A_{13} \\ A_{31} \end{bmatrix} = \begin{bmatrix} \phi_{11} \\ \phi_{13} \\ \phi_{31} \end{bmatrix}. \quad (13d)$$

where $A_{mn} = A_r(m,n)$, and $\phi_{mn} = \phi(m,n)$.

4. SIMULATION RESULTS AND DISCUSSION

For computer simulations, the input image is divided into 8×8 blocks, and the transform coefficients are quantised with a uniform quantiser using the quantisation step-size table recommended in JPEG [7]. The intensities of corrupted block are replaced with 128, which is a middle value of the image intensity.

In Fig. 3, the PSNR performance according to the BLR on the *Lena* image are shown. Even at the high BLR as 10^{-1}, it is found that the recovered images show high PSNR values. With the BLR of $O(10^{-2})$, the recovered images are hardly distinguishable from their error-free images. In Fig. 4, the corrupted *Lena* image with a BLR of 5×10^{-2} is shown. And, the Fig. 5 shows the reconstructed image of Fig. 4 by the proposed algorithm. Except for the blocks having sharp edge or in successive damaged area, we can see that the reconstructed blocks are hardly recognised.

Now we shall briefly discuss the computational complexity. The proposed algorithm requires fewer multiplications and additions, compared to other EC algorithms, such as [1], [4]. It can be shown that the computational complexity of [1] is $O(N^3)$. Also note that [4] is based on the complex fussy logic computations. On the other hand, the numbers of multiplications and additions of the proposed algorithm can be obtained by examining (7) and (12), which are $4\nu(4N+\nu)$, and $4\nu(8N+\nu-2)$, respectively, For example, in the case of $N = 8$, and $\nu = 3$, the number of multiplications and additions are 420 and 780, respectively.

5. CONCLUSION

A new EC technique for DCT based image coding has been presented in this paper. The proposed algorithm recovers the damaged blocks based on the smoothness property of an image at the boundaries of the blocks. In our approach, the intersample variations between adjacent blocks are described by an object function, and the DCT coefficients which minimise the object function are obtained by solving a linear equation. Furthermore, we have shown that the linear equation could be decomposed and reduced into more simple sub-equations. Thus, the computational complexity of the proposed algorithm is very low, compared to the existing techniques. The simulation results indicate that the proposed algorithm recovers the damaged blocks, even if the BLR is as high as order of 10^{-2}. Further employment of the proposed algorithm includes ATM layered coding and full-motion video applications, such as the H.261 and MPEG.

REFERENCES

[1] Y. Wang, Q. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Trans. Commun.*, vol. COM-41, no. 10, pp. 1544-1551, Oct. 1993.

[2] H. Sun and J. Zdepski, "Adaptive Error Concealment Algorithm for MPEG Compressed Video," *Proc. SPIE Int. Conf. Visual Commun. and Image Proc.*, Vol. 1818, Boston, Nov. 1992. pp. 814-824.

[3] M. Ghanbari, and V. Seferidis, "Cell-loss concealment in ATM video codes," *IEEE J. Video Technology*, vol. 3, no. 3, pp. 248-258, June, 1993.

[4] X. Lee, Y. Zhang, and A. Leon-Garcia, "Information loss recovery for block-based image coding techniques - A fussy logic approach," it Proc. SPIE int. Conf. Visual Commun. and Image Proc., Vol. 2094, Cambridge, Nov. 1993. pp. 529-540

[5] Q. Zhu, W. Wang, and L. Shaw, "Joint Source Coding and Packetisation for Video Transmission over ATM Networks," *Proc. SPIE Int. Conf. Visual Commun. and Image Proc.*, Vol. 1818, Boston, Nov. 1992. pp. 837-848.

[6] S. K. Chan, and A. Leon-Garcia, "Block loss for ATM video," it Proc. SPIE int. Conf. Visual Commun. and Image Proc., Vol. 2094, Cambridge, Nov. 1993. pp. 213-222

[7] Joint Photographic Experts Group, ISO/IEC/JTC1/SC2/WG8, "JPEG technical specification, Revision 8," JPEG-8-R8, Aug., 1990.

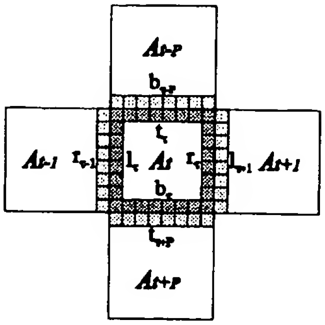


Figure 1. The relations of boundary vectors in object function.

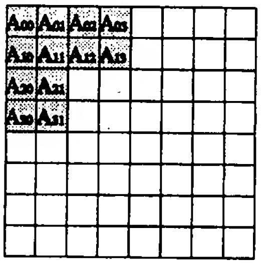

A_{00} A_{01} A_{02} A_{03}
A_{10} A_{11} A_{12} A_{13}
A_{20} A_{21}
A_{30} A_{31}

Figure 2. The selected 12 coefficients in (14).

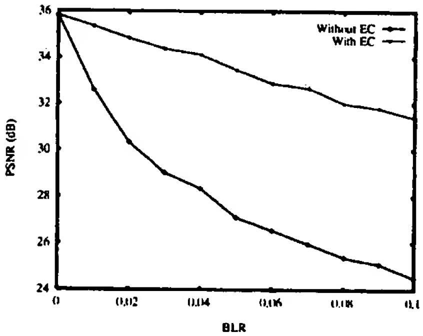


Figure 3. Performance (PSNR) with respect to block loss rate.

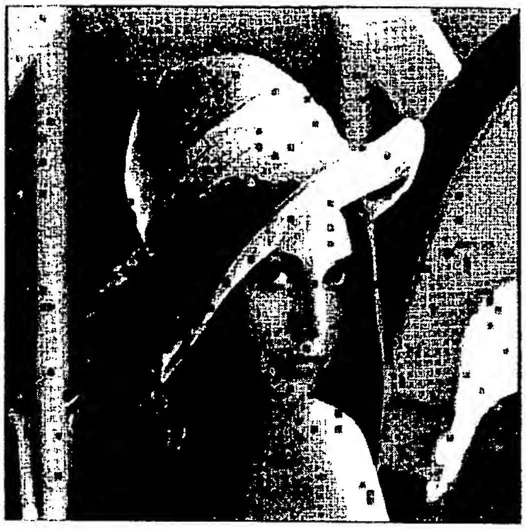


Figure 4. Corrupted *Lena* image with a BLR of 5×10^{-2}.

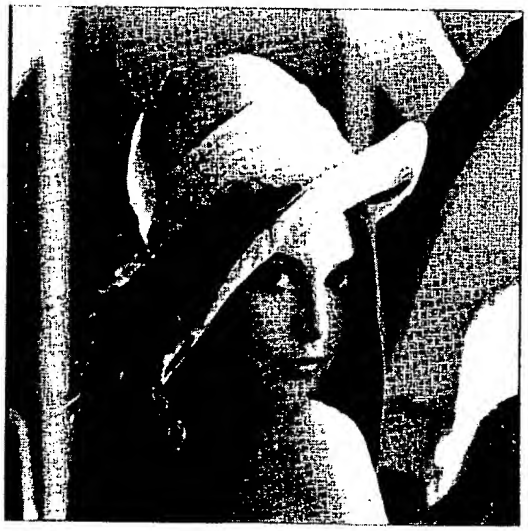


Figure 5. Reconstructed image of Fig. 4 by the proposed algorithm.

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.5

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review Quick Links » Abstract Plus

Welcome to IEEE Xplore®
- ○ Home
- ○ What Can I Access?
- ○ Log-out

Tables of Contents
- ○ Journals & Magazines
- ○ Conference Proceedings
- ○ Standards

Search
- ○ By Author
- ○ Basic
- ○ Advanced

Member Services
- ○ Join IEEE
- ○ Establish IEEE Web Account
- ○ Access the IEEE Member Digital Library

Print Format

SEARCH RESULTS [PDF Full-Text (368 KB)] PREVIOUS NEXT DOWNLOAD CITATION

A survey of the state of the art and utilization of embedded, tree-based coding

Pearlman, W.A. Said, A.

Dept. of Electr. Comput. & Syst. Eng., Rensselaer Polytech. Inst., Troy, NY;

This paper appears in: **Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on**
Meeting Date: 05/31/1998 -06/03/1998
Publication Date: 31 May-3 Jun 1998
Location: Monterey, CA , USA
On page(s): 114-117 vol.5
Volume: 5, References Cited: 33
Number of Pages: 6 vol. (xlv+603+489+674+615+557+656)
INSPEC Accession Number: 6033964

Abstract:

We review first the development of tree-based, embedded image coding. We then explore their use with different image transformations, the reasons for their effectiveness and low complexity, their flexibility and incorporation into state-of-the-art compression systems

Index Terms:

data compression image coding trees (mathematics) complexity embedded image coding flexibility image transformations state-of-the-art compression systems tree-based coding

Documents that cite this document

Select link to view other documents in the database that cite this one.

SEARCH RESULTS [PDF Full-Text (368 KB)] PREVIOUS NEXT DOWNLOAD CITATION

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms| Back to Top

A SURVEY OF THE STATE-OF-THE-ART AND UTILIZATION OF EMBEDDED, TREE-BASED CODING

William A. Pearlman

Rensselaer Polytechnic Institute
ECSE Dept.
Troy, NY 12180 — pearlman@ecse.rpi.edu

Amir Said

Iterated Systems, Inc.
3525 Piedmont Road, Suite 7/600
Atlanta, GA 30025 — said@ipl.rpi.edu

ABSTRACT

We review first the development of tree-based, embedded image coding. We then explore their use with different image transformations, the reasons for their effectiveness and low complexity, their flexibility and incorporation into state-of-the-art compression systems.

1. INTRODUCTION

Six years after their introduction, tree-based algorithms for embedded image coding established a firm reputation as some of the most powerful, efficient and versatile methods known for image compression. They are now studied by a very active group of researchers, and this paper presents a short survey of the state-of-the-art in the field.

Interestingly, those compression methods initially faced a certain skepticism. They were able to support very desirable features, like embedded coding, fast compression and decompression, very precise distortion or rate control, support for lossless compression, great scalability, no need for training, etc. At the same time, they had a performance superior to even the methods painfully adapted to support only one (or a few) of those features. For some, it seemed definitely too good to be true.

One possible reason for this skepticism was that they use two techniques that had been used for some time, but which had been yielding mediocre results. The first technique is tree-based recursive partitioning schemes—as employed in quadtree image compression [26]. The second is bit-plane [13], which permits progressive image transmission with gradual quality improvement.

The origins of these methods can be traced back to when subband/wavelet [30, 31] image coding methods began to gather wider acceptance, as even very simple schemes could produce remarkably good visual quality, together with progressive image transmission. Lewis and Knowles [9] were the first to use trees to exploit the statistical properties found in the pyramidal decomposition of natural images. They proposed an efficient decomposition and coding scheme using tree structures that follow the same spatial location across different subbands, which we call spatial orientation trees.

Then, Shapiro [21, 22] proposed a very clever method to combine bit-plane coding, applied to the wavelet coefficients, with a tree-based partitioning similar to the one developed by Lewis and Knowles. This combination, called embedded zerotree of wavelets (EZW), is by no means a trivial juxtaposition of methods. It identified that efficient compression required separately coding the significance data (to identify if a pixel magnitude is larger than a threshold) and the rest of the bit planes. Further, it shows how much tree structures are efficient to code significance data of wavelet image pyramids.

Shortly after the EZW introduction, we presented a more general analysis of the algorithm [14, 15], relating it to some selection (sorting) and set-partitioning problems. We also proposed some changes that improved its performance, and have shown how to use it for lossless image compression [16, 18]. Later we studied how to improve the method to achieve better embedding, faster coding and decoding, simpler and more efficient entropy-coding, and less memory usage. The result was a new algorithm, called SPIHT, that indeed achieved all those goals [17].

In this paper we present a cross section of what happened in the field recently. Due to lack of space the survey is not supposed to be comprehensive, and we apologize for all missing references. Also, we had to exclude important work that is embedded but not tree-based, and tree-based but not embedded, extensions to video, etc.

2. USE WITH DIFFERENT IMAGE TRANSFORMATIONS

One common misconception about tree-based embedded methods is that they work only with wavelets. They were indeed designed to exploit special properties of the wavelet pyramid coefficients, under a particular tree structure. Nonetheless, a number of experiments show that they are quite effective for other transformations as well. For instance, they were sucessfuly employed together with 8 × 8 DCTs, and even better results are obtained with 16 × 16 DCTs, [32, 11].

Among several exciting developments we have schemes for wavelet packets [33] and general lapped block transforms [28]. The lifting scheme [27] allows the development of better transforms for lossless compression [1], and wavelets on manifolds [20]. Tree-based embedded coding methods are readily adapted anywhere the lifting scheme can be used. For example, an interesting application is efficient (and embedded) compression of functions defined on

0-7803-4455-3/98/$10.00 © 1998 IEEE

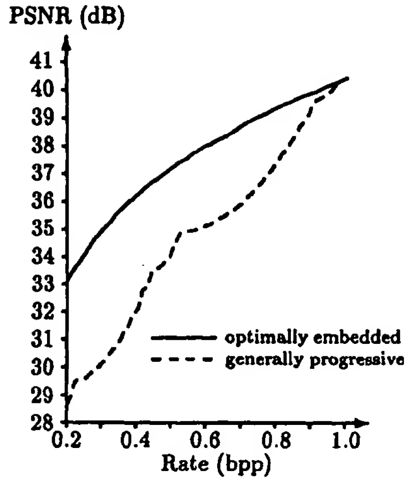


Figure 1: Performance of different preogressive transmission methods when used for embedded coding.

the surface of sphere [6].

While reading this paper the reader should keep in mind that we are discussing a coding algorithm, and as such, its performance (compression efficiency, memory usage, reconstruction visual quality, etc.) depends on the image transformation being used.

3. OPTIMIZED EMBEDDED CODING AND NON-EMBEDDED CODING

Shapiro [22] correctly defined his embedded coding with two objectives: "(1) obtaining the best image quality for a given bit rate; (2) all encodings of the same image at lower bit rates are embedded in the beginning of the bit stream for the target bit rate." Without the first condition, almost any coding method can be called embedded, but possibly with dismal performance at intermediary rates. For example, Figure 1 shows the performance[1] of SPIHT [17], compared to a method designed for progressive resolution [19], adapted for embedded coding.

Unfortunately, now the literature is filled with methods using the expression embedded coding incorrectly. In fact, image pyramids are sometimes called "embedded wavelets," so that any coding method using them is also called embedded.

It is important to observe that bit-plane embedded coding does not exclude the possibility of using other forms of progressive transmission. For instance, with the proper file format it is possible to recover subbands independently. An impressive application based on this fact is a program that allows real-time enhancement—up to lossless recovery—in an user-selected region [4].

Embedded coding necessarily adds some extra complexity to the coding algorithm, as bit planes are sequentially visited, and it is true that embedded coding methods may not be as fast as the fastest non-embedded methods. However, embedded tree-based methods are not much worse either. First, it should be noted the computational effort spent in the first passes is insignificant compared to the effort in the last pass. So, it is not too different from a single-pass algorithm (non tree-based do not have this property). Second, it is not a fair comparison, as there are "shortcuts" that can be taken when embedded coding is not required.

4. THEORETICAL ANALYSIS

The development of tree-based embedded coding methods followed an unusual path: a rigorous theoretical basis is being discovered many years after the practical methods were devised from empirical analysis of distributions in wavelet pyramids formed from natural images. Little was known about which images would be "natural" enough to yield similar results. In the same token, little was known about how to use these methods with different image transformations, like DCTs.

Davis and Chawla [5] proposed a method based on dynamic programming to find the optimal tree structures (set-partitioning rules) for different transformations (like DCTs). Surprisingly, the variation found among different trees can be quite small, showing that tree-based algorithms are by nature quite robust.

New work [25] model the dependence that exist between the magnitudes of wavelet coefficients in different scales, and orientation. This phenomenon originated the use of the expression "self-similarity" to explain the efficacy of tree-based coding. However, it was never used in [21, 22, 15, 17] to mean fractal self-similarity: it meant similarity in a distribution sense. So, in contrast with theories based on conceptual analysis [8], the statistical models above provide not only an explanation to the success of tree-based coding with wavelets, but also to other transforms.

5. COMPUTATIONAL COMPLEXITY

The speed of bit-plane embedded coding methods is defined by the need to perform several passes on the image, comparing magnitudes each pass. The tree-based methods have an advantage over other methods because the necessary information is stored in linked lists, which are, for the most common bit-rates, much smaller than the total number of pixels. Experiments show that most of the CPU time is spent testing for significance, and a small fraction is used for pixel value refinement.

The encoder needs to first compute the maximum magnitudes of the pixels in the trees, but this can be done with very efficiently with a single pass of binary-OR operations done during the image transformation [23]. So, this pass through the whole image is computationally equivalent to a single pass of other bit-plane algorithms.

[1] PSNR = $10\log_{10}(255^2/(\text{mean squared error}))$.

Next, both the encoder and decoder have to maintain a dynamic list of coefficients. Shapiro's EZW algorithm [22] adds four descendants to the lists whenever a parent or descendant is significant. One improvement is to add descendants only when at least one descendant is significant [14, 15]. This yields a very significant reduction in the number of elements in the lists, with proportional reduction in CPU time and memory required for list processing.

The next speed bottleneck was the requirement of using arithmetic coding for effective compression. The following breakthrough was the proposal of a different—and even more compact—tree set partitioning [17], which would keep more insignificant descendants grouped together. This representation was efficient enough to yield excellent results, even without any form of entropy-coding. Furthermore, it allows dealing with groups of 2 × 2 pixels/trees, for another significant reduction of list management resources.

The first implementations had nearly equal encoding and decoding times, but this was due to the arithmetic code. With faster entropy-coding the decoder can be quite faster because it does not require computation of maximum magnitudes.

Those advanced new coding algorithms yielded such large reduction in the coding/decoding complexity, that, in a software implementation, coding/decoding became much faster than the image transformation. Even though it is frequently thought that list processing is a complex task, it is a well known programming problem, and thus has been extensively studied and optimized. Consequently, tree-based decoding algorithms can be faster than any other algorithm, because they process only a small fraction of the image pixels.

Significant improvements have been proposed for specialized architectures, and parallel processing [2]. A method to exploit embedded coding to reduce the complexity of the wavelet forward and reverse transformation is proposed by Paris et al. [12].

6. RESILIENCE TO TRANSMISSION ERRORS

Error control is done by adding redundant bits, or by asking for retransmission. This cause an overhead on the bandwith that depends on the error rates, and the efficiency of the protection method. Transmission errors can drastically reduce the effective bandwidth if improper error control procedures are used. For instance, assuring no data errors all the time is indispensable for text or database data, but it is not necessary for still images and video. This happens because we can still visually gather most of the information if only some parts are degraded. However, in all efficient compression methods an error in the compressed data can lead to a possibly long sequence of errors, or a catastrophic loss of all data after the error.

As explained above, embedded coding is, by definition, based on sorting the image components according to their importance. This can be extremely useful when designing error control systems, as error protection resources can be assigned according to the importance of the data.

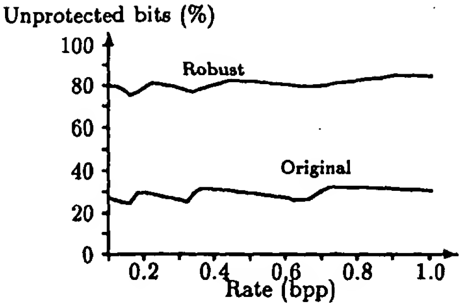


Figure 2: Fraction of bits that do not need error protection, according to embedded coding method.

Bit-plane coding helps in two ways. First, it guarantees that only the data below a certain plane is corrupted after an error. Second, the least significant bits do not require entropy coding. So, for those bits it is not necessary to use error protection because without variable-length coding there is no catastrophic error propagation.

On the other hand, all error in significance data do lead to catastrophic error propagation. The significance data can be protected with common error-correction codes. An interesting alternative it to slightly change the coding algorithm [10] to avoid error propagation. Figure 2 shows the fraction of bits that do not lead to error propagation in a bit-plane compressed file (assuming a previous wavelet decomposition), using the original and modified coding methods.

Several other ingenious solutions have been proposed to improve resilience to transmission errors, like special transmission protocols, adapted algorithms, combination with error correction, etc. [3, 24, 29].

7. CONCLUSIONS

We have traced the development of tree-based, embedded coding techniques to Lewis and Knowles [9] and Shapiro [22] and have shown their current realization in SPIHT coding and several variants to accommodate more features, such as resilience to channel errors and region-of-interest coding. It is apparent that, due to the high performance, low complexity, and versatility of these techniques, they represent the current state-of-the-art in image coding.

8. REFERENCES

[1] R. Calderbank, I. Daubechies, W. Sweldens and B.-L. Yeo, "Lossless image compression using integer to integer wavelet transforms," Proc. IEEE Conf. Image Processing, 1997.

[2] C.D. Creusere, "Image coding using parallel implementations of the embedded zerotree wavelet algorithm," Proc. SPIE, Digital Video Compression: Algorithms and Technologies, vol. 2668, pp. 82–92, March 1996.

[3] C.D. Creusere, "A new method of robust image compression based on the embedded zerotree wavelet algorithm," IEEE Trans. Image Processing, vol. 6, pp. 1436–1442, Oct. 1997.

[4] C.D. Creusere, "Spatially partitioned lossless image compression in an embedded framework," Proc. Asilomar Conf. Signals, Syst.,& Computers, Pacific Grove, CA, Nov. 1997.

[5] G.M. Davis and S. Chawla, "Image coding using optimized significance tree quantization," Proc. Data Compression Conf., pp. 387–396, Snowbird, Utah, March 1997.

[6] K. Kolarov and W. Lynch, "Compression of functions defined on surfaces of 3D objects," Proc. Data Compression Conf., pp. 251–260, Snowbird, Utah, March 1997.

[7] B.-J. Kim and W.A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)," Proc. Data Compression Conf., pp. 251-260, Snowbird, Utah, March 1997.

[8] O. Kiselov and P. Fisher, "Image compression with iterated functions systems, finite automata and zerotrees: grand unification," Proc. Data Compression Conf., p. 443, Snowbird, Utah, March 1996.

[9] A.S. Lewis and G. Knowles, "A 64 Kb/s video codec using the 2-D wavelet transform," Proc. Data Compression Conf., pp. 196–201, Snowbird, Utah, 1991.

[10] H. Man, F. Kossentini, M.J.T. Smith, "Robust EZW Image Coding for Noisy Channels," IEEE Signal Processing Letters, vol. 4(9), pp. 227–229, Aug. 1997.

[11] D.M. Monro and G.J. Dickson, "Zerotree Coding of DCT Coefficients," Proc. IEEE Int. Conf. Image Processing, 1997.

[12] J. Paris, J. Alamanac, R. Mateu, X. Masvidal, and X. Ginesta, "Low bit rate software-only wavelet video coding," IEEE First Workshop Multimedia Signal Processing, pp. 169–174, Princeton, NJ, June 1997.

[13] M. Rabbani, and P.W. Jones, Digital Image Compression Techniques, SPIE Opt. Eng. Press, Bellingham, Washington, 1991.

[14] A. Said, An Improved Zero-Tree Algorithm for Image Compression, Image Processing Lab. Report IPL TR-122, Rensselaer Polytechnic Institute, Troy, NY, Nov. 1992.

[15] A. Said and W.A. Pearlman, "Image compression using the spatial-orientation tree," IEEE Int. Symp. Circuits & Systems, Chicago, IL, pp. 279–282, May, 1993.

[16] A. Said and W.A. Pearlman, "Reversible image compression via multiresolution representation and predictive coding," SPIE Conf. Visual Commun. Image Processing, Proc. SPIE 2094, pp. 664–674, Cambridge, MA, Nov. 1993.

[17] A. Said and W.A. Pearlman, "A new fast and efficient codec based on set partitioning in hierarchical trees," IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 243–250, June 1996.

[18] A. Said and W.A. Pearlman, "An image multiresolution for lossless and lossy compression," IEEE Trans. Image Processing, vol. 5, pp. 1303-1310, Sept. 1996.

[19] A. Said and W.A. Pearlman, "Low-complexity waveform coding via alphabet and sample-set partitioning," Proc. SPIE Conf. Visual Commun. Image Processing, vol. 3024, pp. 25–37, San Jose, CA, Feb. 1997.

[20] P. Schöder and W. Sweldens, "Spherical wavelets: efficiently representing functions on the sphere," Computer Graphics Proc., (SIGGRAPH) pp. 161-172, 1995.

[21] J.M. Shapiro, "An embedded wavelet hierarchical image coder," Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, San Francisco, CA, pp. IV 657–660, March 1992.

[22] J.M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients," IEEE Trans. Signal Processing, vol. 41, pp. 3445–3462, Dec. 1993.

[23] J.M. Shapiro, "A fast technique for identifying zerotrees in the EZW algorithm," Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, vol. 3, pp. 1455–1458, May 1996.

[24] P.G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," IEEE Signal Processing Letters, vol. 4(7), pp. 189–191, July 1997.

[25] E.P. Simoncelli, "Statistical models for images: compression, restoration and synthesis," Asilomar Conf. Signals, Syst., Computers, Pacific Grove, CA, Nov. 1997.

[26] G.J. Sullivan and R.L. Baker, "Efficient quadtree coding of images and video," IEEE Trans. Image Processing, vol. 3, pp. 327–331, May 1994.

[27] W. Sweldens, The Lifting Scheme: A Custom-Design Constructions of Biorthogonal Wavelets, Tech. Paper 1994:7, IMI, Dept. of Mathematics, Univ. of South Carolina, 1994.

[28] T. Tran and T. Nguyen, "A progressive transmission image coder using linear phase paraunitary filter banks," Asilomar Conf. Signals, Syst., Computers, Pacific Grove, CA, Nov. 1997.

[29] S. Thillainthan, D. Bull, and N. Canagarajah, "Robust embedded zerotree wavelet coding algorithm," Proc. SPIE, Visual Commun. Image Processing, San Jose, CA, Jan. 1998.

[30] J.W. Woods and S.D. O'Neil, "Subband coding of images," IEEE Trans. Acoust., Speech, Signal Processing, vol. 34, pp. 1278–1288, Oct. 1986.

[31] J.W. Woods, ed., Subband Image Coding, Kluwer Academic Publishers, Boston, MA, 1991.

[32] Z. Xiong, O.G. Guleryuz and M.T. Orchard, "A DCT-based embedded image coder," IEEE Signal Processing Letters, vol. 3(11), pp. 1289–290, Nov. 1996.

[33] Z. Xiong, K. Ramchandran and M.T. Orchard, "Wavelet packet image coding using space-frequency quantization," submitted to IEEE Trans. Image Processing, 1996.